الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université du 20 Aout 1955-Skikda
Faculté des Sciences
Département d'Informatique

جامعة 20 أوت 1955- سكيكدة
كلية العلوم
قسم الإعلام الآلي

Thèse

Soumis en conformité avec les exigences du diplôme de
3é cycle doctorat (LMD)
Spécialité : Informatique
Option : Systèmes Informatiques

Techniques d'Optimisation et d'Apprentissage pour la Recherche de l'Image par le Contenu Visuel

---

Présentée par :

Boukerma Rahima

Soutenu publiquement le : 27/01/2026

Devant le jury composé de :

Abdelhak Mansoul	MCA	Université 20 Août 1955 - Skikda	Président
Mawloud Mosbah	MCA	Université 20 Août 1955 - Skikda	Examinateur
Said Brahimi	MCA	Université 05 Mai 1945 - Guelma	Examinateur
Bachir Boucheham	Professeur	Université 20 Août 1955 - Skikda	Encadreur
Salah Bougueroua	MCB	Université 20 Août 1955 - Skikda	Co-Encadreur

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

University of Skikda- 20 Août 1955
Faculty of Sciences
Department of Computer Science

جامعة 20 أوت 1955- سكيكدة
كلية العلوم
قسم الإعلام الآلي

Thesis

Submitted in partial fulfillment of the requirements for the degree of
3rd cycle doctorate (LMD system)
Major: Computer Science
Minor: Computer Systems

Optimization and Learning Techniques for Content-Based Image Retrieval (CBIR)

Publicly defended by:

Rahima Boukerma

On *January 27, 2026*

Examination Committee:

Abdelhak Mansoul	MCA	University of Skikda - 20 Août 1955	President
Mawloud Mosbah	MCA	University of Skikda - 20 Août 1955	Examiner
Said Brahimi	MCA	University of Guelma - 05 Mai 1945	Examiner
Bachir Boucheham	Professor	University of Skikda - 20 Août 1955	Supervisor
Salah Bougueroua	MCB	University of Skikda - 20 Août 1955	Co-supervisor

Dedicated to my parents

Acknowledgments

First and foremost, all praise and thanks are due to Allah who granted me strength, guidance and patience to carry out this research.

I would like to express my profound gratitude to my supervisor, Prof. Bachir Boucheham, for having trusted in my abilities and offering me this interesting research subject. His expertise, guidance, constant support and, above all, his encouragement helped me to stay focused and motivated throughout the course of this thesis. I am also deeply grateful to my co-supervisor, Dr. Salah Bougueroua, for his availability, encouragement, valuable insights and continuous guidance that have enriched the quality of this thesis.

I extend my sincere gratitude to each member of the esteemed examination committee. It is with great appreciation that I thank Dr. Abdelhak Mansoul for honoring me by presiding over the jury. Many thanks and sincere appreciation are also due to Dr. Mawloud Mosbah and Dr. Said Brahimi for their time, thorough review and careful evaluation.

Special thanks are due to the staff of the computer science department for their cooperation, availability and kind assistance during the course of my Ph.D. study.

Finally, my heartfelt thanks go to my parents for their sacrifices, patience and endless support throughout my academic journey.

“Nothing is more important than understanding the nature of things.”

— Mohammed ibn Musa al-Khwarizmi
(780–850)

Contents

Acknowledgments i

Contents iii

List of Figures viii

List of Tables x

Abbreviations xii

ملخص xiv

Abstract xvi

Résumé xviii

Part I: Thesis Overview **1**

Chapter 1 Introduction **2**

1.1 Background and motivation 2

1.2 Aims and objectives 4

1.3 Contributions 5

1.3.1 Developing discriminative and robust feature representations 5

1.3.2 Developing effective feature dimensionality reduction methods 6

1.4 Thesis outline 8

Part II: State of the Art of CBIR **10**

Chapter 2 Content-Based Image Retrieval **11**

2.1 Introduction 11

2.2 A broad view of image retrieval systems 11

2.2.1 Information retrieval……. 11

2.2.2 Image retrieval systems 12

2.2.2.1 Text-Based Image Retrieval 13

2.2.2.2 Content-Based Image Retrieval 14

2.2.2.3 Semantic-Based Image Retrieval 15

2.3 General framework of a CBIR System 17

2.4 Feature Extraction 18

2.4.1 Handcrafted feature extraction 19

2.4.1.1 Color features 19
2.4.1.2 Texture features 25
2.4.1.3 Shape features 33
2.4.1.4 Spatial information features 35
2.4.2 Learning-based feature extraction 36
2.5 Matching Measures 37
2.5.1 Euclidean distance 38
2.5.2 Manhattan distance 38
2.5.3 Chebyshev distance 38
2.5.4 Canberra distance 39
2.5.5 Square chord distance 39
2.5.6 D1 distance 40
2.6 Performance evaluation 40
2.6.1 Precision 41
2.6.2 Recall 41
2.6.3 F-Score 42
2.7 Benchmark datasets 43
2.7.1 General-purpose image datasets 43
2.7.2 Domain-specific image datasets 44
2.8 Practical applications of CBIR 49
2.9 Conclusion 53

Chapter 3 Local Patterns-Based Methods for Image Characterization 54

3.1 Introduction 54
3.2 Overview 54
3.2.1 Advantages of local pattern methods 55
3.2.2 Limitations of local pattern methods 55
3.3 Local Binary Pattern 56
3.4 LBP variants 59
3.4.1 Grayscale-based LBP variants 59
3.4.1.1 Dominant LBP 59
3.4.1.2 Improved LBP 60
3.4.1.3 Local Ternary Patterns (LTP) 61
3.4.1.4 Modified LBP 63
3.4.2 Color-based LBP variants 63

3.4.2.1 Opponent Color LBP 64
3.4.2.2 Improved Opponent Color LBP 65
3.4.2.3 Multichannel adder-based and decoder-based LBPs 66
3.4.2.4 Multi-channel local ternary pattern 67
3.4.3 LBP-based hybrid methods 69
3.4.3.1 Combining different LBP variants 70
3.4.3.2 Combing LBP with other handcrafted methods 70
3.4.3.3 Combining LBP with deep learning methods 71
3.5 LBP-based feature selection 73
3.5.1 Rule-based selection…… 73
3.5.2 Boosting LBP features.. 74
3.5.3 LBP subspace learning……75
3.5.4 Other methods……. 76
3.6 LBP-based feature weighting 76
3.6.1 Feature weighting concept 76
3.6.2 Weighting techniques for local pattern methods 77
3.7 Conclusion 78

Chapter 4 Optimization and Machine Learning Techniques for CBIR 80

4.1 Introduction 80
4.2 How do optimization and machine learning techniques contribute to CBIR systems enhancement? 80
4.3 Optimization techniques 83
4.3.1 Heuristic methods…… 84
4.3.2 Metaheuristic methods…. 85
4.3.2.1 Evolutionary-based algorithms 87
4.3.2.2 Swarm intelligence-based algorithms 90
4.3.2.3 Human-based algorithms 91
4.3.2.4 Science-based algorithms 93
4.4 Machine learning techniques 96
4.4.1 Unsupervised learning 96
4.4.1.1 K-Means algorithm…….. 97
4.4.1.2 Principal Component Analysis……. 98
4.4.2 Supervised learning……. 98
4.4.2.1 K-Nearest Neighbors……. 99

4.4.3 Deep learning……. 100
4.4.3.1 Convolutional Neural Network 101
4.4.3.2 Autoencoders 102
4.5 Conclusion 104

Part III: Personal Contributions **106**

Chapter 5 Developing Discriminative and Robust Feature Representations: New methods for effective CBIR **107**

5.1 Introduction 107
5.2 O-OCLBP and O-IOCLBP: new color-texture descriptors 108
5.2.1 Overview……. 108
5.2.2 Proposed methodology……. 109
5.2.3 Experimental evaluation 113
5.2.3.1 Experimental details 113
5.2.3.2 Results and analysis 114
5.3 Dynamic pattern weighting approach for enhancing feature representations 122
5.3.1 Overview……. 122
5.3.2 Proposed methodology……. 123
5.3.3 Experimental evaluation 126
5.3.3.1 Experimental setup 126
5.3.3.2 Parameters setting and dataset 126
5.3.3.3 Results and analysis 126
5.4 Conclusion 128

Chapter 6 Reducing Feature Dimensionality: Novel Feature Selection Methods for Effective and Efficient CBIR **129**

6.1 Introduction 129
6.2 A dimensionality reduction method based on particle swarm optimization algorithm 130
6.2.1 Overview……. 130
6.2.2 Proposed methodology……. 131
6.2.3 Experimental evaluation 133
6.2.3.1 Experimental details 133
6.2.3.2 Results and analysis 133
6.3 An enhanced pattern selection scheme based on sine cosine algorithm as a tool for dimensionality reduction 135

6.3.1 Overview……. 135
6.3.2 Proposed methodology……. 136
6.3.3 Experimental evaluation 139
6.3.3.1 Experimental details..... 139
6.3.3.2 Results and analysis 140
6.4 Optimized feature space based on deep learning and sine cosine algorithm..... 146
6.4.1 Overview……. 146
6.4.2 Proposed methodology.. 147
6.4.3 Experimental evaluation 152
6.4.3.1 Experimental details..... 152
6.4.3.2 Results and analysis 153
6.5 Conclusion 155

General conclusions and perspectives **156**

Conclusions..... 156
Perspectives 158

Bibliography **159**

Appendix: Candidate List of Publications **183**

List of Figures

2.1 An example of a text query from Google search engine 16

2.2 General framework of a Content-based image retrieval system 18

2.3 RGB space (Tyagi, 2017) 20

2.4 HSV space (Srivastava et al, 2023) 21

2.5 YCbCr space (Khediri et al, 2021) 21

2.6 CIELAB space (Khediri et al, 2021) 22

2.7 An example of two different images with similar color histograms: (a) Guinean flag, (b) Malian flag, (c) RGB histogram of the Guinean flag, (d) GB histogram of the Malian flag 23

2.8 Example of texture types: (a) natural textured surfaces, (b) artificial textured surfaces (Srivastava et al, 2023) 26

2.9 Boundary-based and region-based representations (Khokher & Talwar, 2012) 34

2.10 Sample images from general-purpose image datasets: (a) Corel dataset, (b) Flickr dataset 44

2.11 Sample images from texture datasets: (a) Brodatz, (b) KTH-TIPS, (c) KTH-TIPS2b, (d) STex, (e) USPTex, (f) BarkTex, (g) NewBarkTex 46

2.12 Sample images from object image datasets: (a) COIL-100, (b) Caltech-256 47

2.13 Samples from medical image datasets: (a) OASIS-MRI, (b) KIMIA Path960, (c) Kather-5K 48

3.1 An example of the original LBP operator construction: (a) 3 × 3 neighborhood, (b) binary pattern, (c) weights, (d) LBP code (Takala et al, 2005) 57

3.2 An example of local ternary pattern operator (adapted from (Pietikäinen & Zhao, 2015)) 62

3.3 OCLBP code computation for: (a) RR color channel pair, (b) GG color channel pair, (c) BB color channel pair, (d) RG color channel pair, (e) RB color channel pair, (f) GB color channel pair (Boukerma et al, 2025a) 65

3.4 Computation of maLBP and mdLBP of an RGB image (Dubey et al, 2016) 67

4.1 Classification of metaheuristic algorithms based on the inspiration source 87

4.2 An illustration of the K-Nearest Neighbors (KNN) algorithm (Sultani & Yousif, 2018) 99

4 .3 Illustration of a typical convolutional neural network (adapted from (Hadid et al, 2023) 102

5.1 Sampling scheme of the construction of the O-OCLBP operator for the RG pair of color channels with an 8-pixel neighborhood 110

5.2 Illustration of the O-OCLBP code construction in the RGB color space, for: (a) the RR pair, (b) GG pair, (c) BB pair, (d) RG pair, (e) RB pair, (f) GB pair (Boukerma et al, 2025a) 110

5.3 Structure of the 8 neighbors of a pixel 111

5.4 Example of a query image from Corel-1K database using: (a) color LBP, (b) OCLBP and (c) O-OCLBP (Boukerma et al, 2025a) 119

5.5 Example of a query image from NewBarkTex database using: (a) color ILBP, (b) IOCLBP and (c) O-IOCLBP (Boukerma et al, 2025a) 119

5.6 Architecture of the proposed dynamic weighting approach 125

6.1 Proposed CBHIR architecture (Boukerma et al, 2023) 131

6.2 Comparison of the average retrieval time between the reduced and full-dimensional features (Boukerma et al, 2023) 135

6.3 Comparative chart of the average retrieval time: (a) KIMIA Path 960, (b) Kather-5K (Boukerma et al, 2025b) 143

6.4 Proposed CBHIR system for colon images (Boukerma et al, 2024) 148

6.5 Comparison of ResNet18 architecture before (a) and after (b) the application of transfer learning 150

6.6 Comparison of GoogleNet architecture before (a) and after (b) the application of transfer learning 151

6.7 Example of a query images from Kather-5k database, using: (a) the full set deep features, (b) the optimized deep features (Boukerma et al, 2024) 155

List of Tables

3.1 Summary of some LBP variants 68

3.2 Some LBP-based hybrid methods used in the context of CBIR 72

4.1 Summary table of some key metaheuristic algorithms 95

4.2 Summary table of additional learning methods 103

5.1 Performance comparison using various distance metrics on Corel-1k dataset (Boukerma et al, 2025a) 115

5.2 Average precision rates computed in RGB color space using Manhattan distance (Boukerma et al, 2025a) 116

5.3 Average recall rates computed in RGB color space using Manhattan distance (Boukerma et al, 2025a) 116

5.4 Average precision rates computed in different color spaces on Corel-1K and KTH-TIPS2b databases, using Manhattan distance (Boukerma et al, 2025a) 117

5.5 Average precision rates computed in different color spaces on Corel-1K and KTH-TIPS2b databases, using Manhattan distance (Boukerma et al, 2025a) 117

5.6 Comparison of the performance between the proposed descriptors and some state-of-art methods on Corel-1K database (Boukerma et al, 2025a) 120

5.7 Comparison of the performance between the proposed descriptors (in both absence and presence of noise) and some state-of-art-descriptors for the top 100 images (Boukerma et al, 2025a) 121

5.8 Comparison by class of the average precision rates between LBP, WLBP and DWLBP for the top 20 retrieved images (Boukerma et al, 2025a) 126

5.9 Comparative table of the average precision rates between the baseline, weighted and dynamic weighted methods for the 20 retrieved images (Boukerma et al, 2025a) 127

6.1 Performance and feature size comparison for the top 20 images using Manhattan distance (Boukerma et al, 2023) 134

6.2 Retrieval performance on Kimia Path960 database for the top 20 images (Boukerma et al, 2025b) 140

6.3 Retrieval performance on Kather-5K database for the top 20 images (Boukerma et al, 2025b) 141

6.4 Number of features on Kimia Path960 database for the top 20 images when considering the APR as fitness function (Boukerma et al, 2025b) 142

6.5 Number of features on Kather-5K database for the top 20 images when considering the APR as fitness function (Boukerma et al, 2025b) 142

6.6 Comparison of the feature dimension and the retrieval performance between the proposed methods and some deep learning comparison ng-based methods (Boukerma et al, 2025b) 144

6.7 Comparison of the average precision rates (%) between the proposed method and some state-of-the-art methods (Boukerma et al, 2025b) 145

6.8 Comparison of the feature dimension and performance for the top 20 images (Boukerma et al, 2024) 153

6.9 Comparative table of the average retrieval time (seconds) (Boukerma et al, 2024) .. 154

6.10 Comparison of the average precision rates between the proposed and some state-of-the-art methods (Boukerma et al, 2024) 154

Abbreviations

AE	Autoencoder
ANN	Artificial Neural Network
APR	Average Precision Rate
ARR	Average Recall Rate
BBO	Biogeography-Based Optimization
BoF	Bag of Features
CAD	Computer-Aided Diagnostic
CBIR	Content-Based Image Retrieval
CBHIR	Content-Based Histopathology Image Retrieval
CBMIR	Content-Based Medical Image Retrieval
CNN	Convolutional Neural Network
CS	Cuckoo Search
DE	Differential Evolution
DL	Deep learning
DLBP	Dominant LBP
DPW	Dynamic Pattern Weighting
DT	Decision Tree
EA	Evolutionary Algorithm
FS	Feature Selection
FW	Feature Weighting
GA	Genetic Algorithm
GLCM	Gray-Level Co-occurrence Matrix
HOG	Histogram of Oriented Gradients
ILBP	Improved LBP

IOCLBP	Improved Opponent Color Local Binary Patterns
IR	Image retrieval
KNN	K-Nearest Neighbors
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LTP	Local Ternary Patterns
ML	Machine Learning
maLBP	multichannel adder-based LBP
mdLBP	multichannel decoder-based LBP
MLBP	Modified LBP
MRI	Magnetic Resonance Imaging
NFL	No-Free-Lunch
OCLBP	Opponent Color Local Binary Patterns
O-IOCLBP	Orthogonal IOCLBP
O-OCLBP	Orthogonal OCLBP
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
RF	Relevance Feedback
SA	Simulated Annealing
SBIR	Semantic-Based Image Retrieval
SCA	Sine Cosine Algorithm
SVM	Support vector machine
TBIR	Text-Based Image Retrieval
TLBO	Teaching–Learning-Based Optimization
ULBP	Uniform LBP

ملخص

في ظل التقدم التكنولوجي المتسارع المصحوب بالارتفاع الهائل لحجم قواعد بيانات الصور الرقمية، تمكن نظام "البحث عن الصور بناء على المحتوى" أو ما يعرف اختصارا بـ CBIR من اكتساب اهمية متزايدة في العديد من المجالات المرتبطة بالرؤية الحاسوبية. الاهتمام المتزايد بمجال البحث عن الصور بناء على المحتوى يعود الى قدرته على توفير أدوات قوية لتنظيم و هيكلة هذه الصور و البحث عنها في قواعد البيانات الضخمة. يعتمد CBIR أساسا على استعمال المحتوى المرئي للصور مثل اللون، النسيج و الشكل لاستخراج سمات تمييزية قادرة على تمثيل الصور بفعالية. يعد اللون و النسيج من السمات المستعملة بكثرة في CBIR نظرا لقوة التمييز التي تتمتع بها هذه السمات.

بذلت الاوساط البحثية في مجال CBIR جهودا حثيثة و مستمرة على مدار السنوات من أجل جعل هذا المجال اكثر فعالية و كفاءة. تركزت هذه المجهودات بشكل رئيسي على سد ما يعرف ب "الفجوة الدلالية" الناتجة عن الفارق و التباين بين السمات المنخفضة المستوى المستخرجة آليا من الصور، من جهة، و المفاهيم او المعاني الدلالية عالية المستوى التي يدركها الانسان و تعكس فهمه لهذه الصور، من جهة أخرى. مشكلة الابعاد المرتفعة او ما يعرف ب "لعنة الأبعاد" هي مشكلة أخرى مهمة في مجال البحث عن الصور بناء على المحتوى، التي تنشأ عند التعامل مع فضاءات سمات ذات ابعاد عالية. تحدي آخر ذو أهمية بالغة في هذا المجال، يكمن في تطوير طرق فعالة لتمثيل سمات الصور. في هذا السياق، أثبتت تقنيات التحسين و تعلم الآلة فعالية كبيرة من الناحية العمليّة لحل مشاكل البحث عن الصور بناء على المحتوى.

و من هذا المنطلق، نسعى في هذه الأطروحة الى تقديم عدة طرق لمعالجة التحديات الاساسية التي تواجه أنظمة البحث عن الصور بناء على المحتوى مع الاستغلال الامثل لتقنيات التحسين و تعلم الآلة. حيث نقوم بشكل رئيسي بـ: (1) تطوير تمثيلات تمييزية لسمات الصور بالاعتماد على محتويات اللون و النسيج، بالإضافة الى تمثيلات اخرى محسنة باستخدام تقنيات التحسين. و هذا بهذف استخراج سمات متينة و مميزة تساهم في تقليص الفجوة الدلالية. (2) استعمال طرق تعلم الآلة بالموازاة مع تقنيات التحسين لتقليل السمات ذات الابعاد العالية. و تجدر الاشارة الى انه في كلتا الحالتين، نستعمل خوارزميات ميتاهيوريستيك مختارة بعناية.

على وجه التحديد، نقترح واصفان جديدان لاستخراج سمات يدوية متينة و فعالة، تحت اسم: (أ) "OCLBP-متعامد" (O-OCLBP) و (ب) "IOCLBP-متعامد" (O-IOCLBP). الواصفان المقترحان هما نسختان مطورتان للواصفان: الأنماط الثنائية المحلية للألوان المتعاكسة (OCLBP) و الأنماط الثنائية المحلية للألوان المتعاكسة المحسن (IOCLBP)، على التوالي. في كلتا الطريقتين المقترحتين، نستعمل آلية الالوان المتعاكسة المتعامد عوضا عن الطوبولوجيا الدائرية المستخدمة في طرق OCLBP و IOCLBP. علاوة على ذلك، نقدم اسهاما آخرا يعتمد على آلية تحسين تعمل على ترجيح ديناميكي لأنماط الصور المستخرجة بواسطة عدة واصفات نسيجية تشمل النمط الثنائي المحلي (LBP)، النمط الثنائي المحلي المحسن (ILBP)، الأنماط الثلاثية المحلية (LTP) والنمط الثنائي المحلي المعدل (MLBP). بصورة ادق، نقترح طريقة اخرى (ج)، تحت اسم "*الترجيح الديناميكي للأنماط* (DPW)" والتي تستعمل

خوارزمية التطور التفاضلي لاستنباط متجه أمثل للأوزان لكل فئة من صور قاعدة البيانات، و هذا لاستعماله لترجيح انماط الصور.

من جهة أخرى، نقترح ثلاث طرق لتقليل السمات ذات الابعاد العالية المستخرجة من صور الانسجة المرضية . (د) نستعمل تقنية تحسينية تعتمد على خوارزمية تحسين سرب الجسيمات (PSO) من اجل تقليل بعد السمات المستخرجة من فضاءات الوان متعددة ومدمجة في متجه سمات متين لكنه ذو بعد عالي. (ه) في تطوير لاحق لهذه المساهمة، تقوم بتوسيع الطريقة السابقة باستعمال تقنية تحسينية بالاعتماد على خوارزمية الجيب وجيب التمام (SCA) لتقليل بعد السمات المستخرجة عن طريق سمات اخرى، تحديدا Combined-LBPs و Combined-OCLBPs. كما نستعمل ايضا قاعدة بيانات اخرى لصور الانسجة المرضية لتقييم أكثر تعقما لطريقة تقليل الأبعاد. و) في الاخير و في عمل آخر، نطبق التعلم بالتحويل على الشبكات العصبية التقليدية المُدرَّبة مسبقاً و المتمثلة في: ResNet18 و GoogLeNet، و هذا من اجل استخراج سمات عالية المستوى. ثم بعد ذلك، نستعمل آلية لتقليل الابعاد يعتمد على خوارزمية SCA لتحسين هذه السمات المستخرجة بناء على التعلم.

بشكل عام، يمكن القول أن كل الطرق المقترحة ضمن هده الاطروحة تشكل تقدما هاما في معالجة المشاكل الاساسية التي تواجه مجال البحث عن الصور بناء على المحتوى. و من الجدير بالذكر ان النتائج المتحصل عليها في جميع الطرق المقترحة أظهرت أداء متفوقا مقارنة مع بعض الطرق المعتمدة في دراسات سابقة.

الكلمات المفتاحية: البحث عن الصور بناء على المحتوى، التعلم العميق، تقليل الابعاد، استخراج السمات، تقنيات التعلم، ميتاهيوريستيك، واصفات الانماط المحلية، تقنيات التحسين، النسيج

Abstract

With the exponential growth in the size of digital image databases, Content-Based Image Retrieval (CBIR) has gained increasing importance in many computer vision related fields. The growing interest shown for CBIR is due to its ability to provide powerful tools for managing and searching images from large-scale image databases. CBIR is principally based on the use of images′ visual contents, such as color, texture and shape, to extract discriminant features that can represent images effectively. In particular, color and texture features are widely used in CBIR due to their discrimination power.

Over the years, the research community in CBIR has been investing extensive efforts to make CBIR systems more and more effective and efficient. These efforts have been mainly concentrated on bridging what is called the “semantic gap”. The latter is due to the difference between the low-level features extracted from images and the high-level semantic concepts. The curse of dimensionality is another important problem which arises when dealing with high-dimensional feature spaces. Developing effective image feature representations is also a significant challenge in CBIR. In this context, optimization and learning techniques have proven to be practically effective for solving CBIR problems.

In this regard, we put forward in this thesis several methods to address the main challenges of CBIR while making the most of optimization and learning techniques. Mainly, (i) we develop discriminative feature representations based on color and texture contents of images, as well as further representations enhanced through optimization techniques. That is for extracting robust and discriminative features, and thereby reducing the semantic gap. (ii) We also use machine learning methods along with optimization techniques to reduce high-dimensional features. In both cases, carefully chosen metaheuristic algorithms are used.

Specifically, we propose two new image descriptors to extract robust and effective handcrafted features, named (a) Orthogonal OCLBP (O-OCLBP) and (b) Orthogonal IOCLBP (O-IOCLBP). The proposed descriptors are respectively two enhanced versions of the Opponent Color Local Binary Patterns (OCLBP) and the Improved Opponent Color Local Binary Patterns (IOCLBP) color-texture descriptors. In both proposed methods, we use an orthogonal opponent color scheme instead of the circular topology used in the OCLBP and IOCLBP methods. Furthermore, we introduce another contribution that relies on an

optimization mechanism for dynamically weighting the image patterns extracted through several texture descriptors, including Local Binary Pattern (LBP), Improved LBP (ILBP), Local Ternary Patterns (LTP) and Modified LBP (MLBP). More precisely, the proposed method, named (c) Dynamic Pattern Weighting (DPW), uses the Differential Evolution (DE) algorithm to generate an optimal weight vector for each class of the image database to be used for weighting the image patterns. On the other hand, we propose three approaches for reducing the high-dimensional features extracted from histopathological images. (d) We use an optimization technique based on Particle Swarm Optimization (PSO) algorithm to reduce the dimension of the ILBP features extracted from multiple color spaces and combined in a robust feature vector, yet high dimensional. e) In a further development of this contribution, we extend the previous approach by using an optimization technique based on Sine Cosine Algorithm (SCA) to reduce the dimension of other color-texture features, namely Combined-LBPs and Combined-OCLBPs. We also use an additional histopathology image dataset for a more thorough assessment of the dimensionality reduction method. f) Finally, in another work, we apply transfer learning on the pre-trained ResNet18 and GoogLeNet convolutional neural networks to extract high-level features. Then, we use a dimensionality reduction scheme based on SCA to optimize these learning-based features.

Overall, all the proposed methods mark a significant advancement in overcoming the key issues in CBIR. Notably, the obtained results demonstrate superior performance compared to some existing methods.

Keywords: CBIR, deep learning, dimensionality reduction, feature extraction, content-based image retrieval, learning techniques, local pattern descriptors, metaheuristics, optimization techniques, texture

Résumé

Avec la croissance exponentielle de la taille des bases d'images, la recherche d'image par le contenu (CBIR en Anglais) a gagné une importance croissante dans de nombreux domaines liés à la vision par ordinateur. L'intérêt accru envers CBIR est dû à sa capacité à fournir des outils puissants pour la gestion et la recherche des images à partir de bases d'images volumineuses. CBIR est principalement basée sur l'utilisation du contenu visuel des images, comme la couleur, la texture et la forme pour extraire des caractéristiques discriminantes qui peuvent représenter les images efficacement. En particulier, la couleur et la texture sont largement utilisées dans CBIR à cause de leur capacité de discrimination.

Au cours des années, la communauté de recherche en CBIR a investi des efforts énormes pour rendre les systèmes CBIR de plus en plus efficaces et efficients. Ces efforts ont été principalement concentrés sur le comblement de ce que l'on appelle le "fossé sémantique". Ce dernier est dû à l'écart entre les caractéristiques de bas niveau extraites des images et les concepts sémantiques de haut niveau. La malédiction de la dimensionnalité est un autre problème important qui survient lorsqu'on traite des espaces de caractéristiques de haute dimension. Développer des représentations efficaces des caractéristiques de l'image est aussi un défi considérable en CBIR. Dans ce contexte, les techniques d'optimisation et d'apprentissage se sont révélées être pratiquement efficaces pour résoudre les problèmes liés à la recherche d'image par le contenu.

À cet égard, nous présentons dans cette thèse plusieurs méthodes pour résoudre les principaux défis auxquels le CBIR est confronté, tout en profitant des techniques d'optimisation et d'apprentissage. Principalement, (i) Nous développons des représentations discriminantes des caractéristiques reposant sur le contenu de couleur et de texture des images, ainsi que d'autres représentations améliorées par des techniques d'optimisation. Cela vise à extraire des caractéristiques robustes et discriminantes, et ainsi réduire le fossé sémantique. (ii) Nous avons aussi utilisé des méthodes d'apprentissage profond avec des techniques d'optimisation pour réduire les caractéristiques de dimension élevée. Dans les deux cas, des métaheuristiques bien choisies sont utilisées.

Spécifiquement, nous proposons deux descripteurs d'image pour extraire des caractéristiques manuelles robustes et efficaces, appelées (a) Orthogonal OCLBP (O-OCLBP) et (b)

Orthogonal IOCLBP (O-IOCLBP). Les descripteurs proposés sont respectivement deux versions améliorées des descripteurs de couleur-texture: Opponent Color Local Binary Patterns (OCLBP) et Improved Opponent Color Local Binary Patterns (IOCLBP). Dans les deux méthodes proposées, nous utilisons un mécanisme de couleurs opposées orthogonales au lieu de la topologie circulaire utilisée dans les méthodes OCLBP et IOCLBP. De plus, nous introduisons une autre contribution qui s'appuie sur un mécanisme d'optimisation pour pondérer dynamiquement les motifs des images extraites à travers plusieurs descripteurs de texture, y compris Local Binary Pattern (LBP), Improved LBP (ILBP), Local Ternary Patterns (LTP) et Modified LBP (MLBP). Plus précisément, la méthode proposée, appelée (c) Dynamic Pattern Weighting (DPW) utilise l'algorithme de l'évolution différentielle (Differential Evolution, DE) pour générer un vecteur de poids optimal pour chaque classe de la base d'images pour être utilisé pour la pondération des motifs de l'image. D'autre part, nous proposons trois approches pour réduire la dimension élevée des caractéristiques extraites à partir des images histopathologiques. (d) Nous utilisons une technique d'optimisation basée sur l'algorithme d'optimisation par essaim de particules (Particle Swarm Optimization, PSO) pour réduire la dimension des caractéristiques ILBP extraites de multiples espaces de couleur et combinées dans un vecteur de caractéristiques robuste, mais au prix d'une dimension élevée. (e) Dans un développement ultérieur de cette contribution, nous étendons l'approche précédente par l'utilisation d'une technique d'optimisation basée sur l'algorithme sinus-cosinus (Sine Cosine Algorithm, SCA) pour réduire la dimension d'autres caractéristiques de couleur et texture, à savoir LBPs-Combinés et OCLBPs-Combinés. Nous utilisons aussi une base supplémentaire d'images histopathologiques pour une évaluation plus approfondie de la méthode de réduction de dimension. F) Enfin, dans un autre travail, nous appliquons l'apprentissage par transfert sur les réseaux de neurones convolutifs pré-entraînés ResNet18 et GoogLeNet pour extraire des caractéristiques de haut niveau. Par la suite, nous utilisons un mécanisme de réduction de dimensionnalité basé sur SCA pour optimiser ces caractéristiques basées sur l'apprentissage.

Dans l'ensemble, toutes les méthodes proposées marquent un progrès significatif dans la résolution des problèmes clés dans CBIR. Soulignons que les résultats obtenus montrent une performance supérieure comparée à celles de certaines méthodes existantes.

Mots-clés : CBIR, apprentissage profond, réduction de dimensionnalité, extraction des caractéristiques, recherche d'image par le contenu, techniques d'apprentissage, descripteurs de motifs locaux, métaheuristiques, techniques d'optimisation, texture

Part I

Thesis Overview

Chapter 1

Introduction

1.1 Background and motivation

The enormous and continual development of Internet technology, image capturing devices and social networking applications has led to the generation of a massive amount of digital image collections that are increasing day by day. Finding efficient ways to facilitate the management and accessibility of such large image collections have become a challenge for the research community. Image Retrieval (IR) is a major research area that investigates and explores various methods for managing, searching and retrieving images from large image databases.

The early researches on IR, dating back to the late 1970s (Alkhawlani et al, 2015; Jardim et al, 2022), proposed using textual annotations to facilitate the retrieval of similar images. In this context, images in Text-Based Image Retrieval (TBIR) are manually annotated by associating them textual descriptors such as keywords, tags and captioning descriptions. The user utilizes therefore textual queries to retrieve images. In addition, all images of the database are indexed based on the textual descriptors. Although the TBIR approach is useful and appropriate for small image databases with precise textual descriptors, it has some limitations (Srivastava et al, 2023; Unar et al, 2019). The main limitation of TBIR is therefore that it is impractical for large image databases where the manual annotation needs a considerable number of human labors (Alkhawlani et al, 2015; Unar et al, 2019). Another essential limitation is that the manual annotation is often a subjective task (Aboaisha, 2015; Srivastava et al, 2023) . This is because describing images with keywords is dependent on the human perception of the images which may lead to give different descriptions to the same image or an identical description or keyword to different images.

In order to face the limitations imposed by TBIR, Content based image retrieval (CBIR) has emerged as an alternative approach to the textual image retrieval. The term CBIR was originated in 1992 with the work of Kato (Kato, 1992). Since then, CBIR has become an attractive research area, and research in this domain has continued until nowadays. Unlike TBIR, where the images are described and indexed based on the manual annotation, in CBIR, retrieval and indexing tasks are performed automatically based on the visual contents of images (such as color, texture, and shape). Typically, to determine the most similar images to a user query image, the CBIR system extracts automatically the visual features from all images in the database, then compare them to those of the query image using a matching measure. A few years after the emergence of CBIR, many successful systems have been developed. Query by image content (QBIC) (Flickner et al, 1995) is one of the first CBIR systems, that was developed by IBM in the 1990s. This system which is reported as the first commercial version of CBIR systems has greatly influenced the later CBIR systems (Zaheer, 2010), such as Photobook (Pentland et al, 1996), Netra (Ma & Manjunath, 1999), PicHunter (Cox et al, 2000), SIMPLIcity (Wang et al, 2001), Blobworld (Carson et al, 2002), VisualSEEK (Smith & Chang, 1997), MARS (Rui et al, 1997), VISTO (Di Mascio et al, 2010), CTDCIRS (Rao et al, 2011a), and other recent systems such as those proposed in (Fadaei et al, 2017; Gravina et al, 2021; Kashif et al, 2020; Ndung'u et al, 2025; Vieira et al, 2023). Over the years, CBIR systems have gained a remarkable importance in various fields, such as medicine (Kapadia & Paunwala, 2021; Souid et al, 2023), fashion design (Li et al, 2019 ; Varshney et al, 2024), agriculture (Gurubelli et al, 2023; Raja & Karthikeyan, 2022), crime investigation (Li et al, 2021a; Liang et al, 2024), etc. In particular, the medical field has greatly benefited from the potential of CBIR especially with the advancement of digital image acquisition devices which has led to the generation of significant medical image databases. Among the successful CBIR systems that deal with medical images, we quote those proposed in: (Chuctaya et al, 2011; Janati Idrissi et al, 2024; Kumar & Kumar, 2023; Shatnawi et al, 2020; Souid et al, 2023).

Despite the ongoing success of CBIR systems which have usually achieved good retrieval performance, they still have some major challenges on which researchers are still working on constantly. Particularly, the semantic gap (Smeulders et al, 2000) is one of the most challenging obstacles that CBIR confronts and on which researchers are focusing on. Actually, the semantic gap is the difference between the visual information extracted from images through the low-level features and the human perception to these images, which may

result in the irrelevance of the images returned to the user. In other words, it consists of the difference between the semantic meanings of images and their visual contents which are concretely represented by pixels (Duan & Kuo, 2022). Today, the leading research direction of the computer vision community concerns the reduction of the semantic gap which is regarded as the backbone of the current computer vision problems (Duan & Kuo, 2022). Another serious problem that researchers are concerned about is what is known as the "curse of dimensionality" (Hameed et al, 2021). This issue usually occurs when high-dimensional features are extracted from images, and this results in a degradation of the CBIR system performance such as the memory usage, the computation complexity and the accuracy. For this reason, dimensionality reduction has become a critical step in CBIR when dealing with high dimension features (Belarbi et al, 2022).

In addition to the above-mentioned challenges, there are two other key issues that should be considered to ensure an efficient CBIR system. These issues, that have a direct impact on CBIR performance, are the choice of the suitable feature to represent images and the effective metric that measures the distance between images. That is why researchers pay a particular attention to the development of efficient feature descriptors and matching measures. The difficulty of selecting the best image features that best represent the image content is due to the perception subjectivity and the composition complexity of images (Khokher & Talwar, 2012). In this context, the choice of the appropriate feature representations for images plays a crucial role in the success of any CBIR system.

1.2 Aims and objectives

This thesis aims to enhance the content-based image retrieval performance, facing the major challenges in current CBIR systems, especially the semantic gap, the curse of dimensionality and the feature representation. In particular, our study is centered on the investigation of the optimization and learning techniques to develop efficient methods to improve the retrieval performance. The focus should therefore be on introducing novel and original methods based on the optimization and learning techniques to address the above-mentioned issues and consequently enhance the CBIR performance. Developing an accurate and efficient CBIR system primarily requires the development of robust and discriminative feature representations that can reduce of the semantic gap while overcoming the curse of dimensionality problem.

The key objectives of this thesis are summarized as follows:

- Development of robust local feature descriptors able to extract discriminative feature representations from images.
- Optimization of the existing feature representations by using optimization techniques such as metaheuristic algorithms.
- Suggestion of novel feature reduction methods based on either optimization or learning techniques to optimize and reduce high-dimensional feature spaces, especially in medical applications where the features dimension significantly affects the retrieval time.

Overall, achieving discriminative feature representations and reduced feature spaces naturally allows the reduction of the semantic gap and consequently the achievement of our primary goal which is the improvement of the CBIR system performance.

1.3 Contributions

As mentioned in the previous section, the main objective of this thesis is to develop novel and efficient approaches to improve the performance of CBIR systems. For this purpose, our attention is focused basically on developing robust and discriminative feature representations and effective feature dimensionality reduction methods. Thus, our contributions are grouped, in accordance with the above-mentioned objectives, into two categories:

1.3.1 Developing discriminative and robust feature representations

To achieve robust and discriminative feature representations, we first introduced a color-texture descriptor named orthogonal opponent color local binary patterns (O-OCLBP) which is an enhanced version of the opponent color local binary patterns (OCLBP) method (Mäenpää & Pietikäinen, 2004); secondly, we further applied the orthogonal opponent scheme, used for deriving O-OCLBP, on the improved opponent color local binary patterns (IOCLBP) method (Bianconi et al, 2017) to derive another new descriptor that we named orthogonal improved opponent color local binary patterns (O-IOCLBP); furthermore, we proposed a weighting approach named Dynamic Pattern Weighting (DPW) for enhancing the feature representations extracted through some local binary pattern methods. The following briefly describes each of the above-mentioned contributions:

- ***Orthogonal opponent color local binary patterns and orthogonal improved opponent color local binary patterns (Boukerma et al, 2025a)***: In this work we proposed two new color-texture descriptors for content-based image retrieval, namely O-OCLBP and O-IOCLBP. The main challenge from which we have started our approach was to improve the OCLBP descriptor which is a well-known color-texture descriptor based on the concept of "opponent colors". The idea was to suggest an effective yet simple scheme to modify the distribution of the color channels in the neighboring pixels of the OCLBP's inter-channel approach in order to improve its performance and increase its invariance to illumination change. Specifically, to form the inter-channel features of OCLBP, we adopted an orthogonal topology to distribute the couples of color channels between the neighboring pixels instead of the circular topology used in OCLBP. The success of the proposed scheme on OCLBP motivated us to apply it on the improved version of OCLBP (IOCLBP), and this has resulted in a new robust and discriminative descriptor (O-IOCLBP). The effectiveness of the proposed scheme was evaluated using eight image databases, five distances, four color spaces and in presence of noise. The obtained results demonstrated the computational simplicity and good performance of the two proposed descriptors, in addition to their resistance to noise.

- ***Dynamic weighted pattern features (Boukerma et al, 2022)***: To enhance the discrimination power of some local pattern methods such as LBP, ILBP, LTP and MLBP, we introduced a dynamic pattern weighting (DPW) approach in which a dynamic weight vector was associated to each query image feature according to the image class. Thereafter, each value of the weight vector was readjusted according to the importance of the corresponding pattern in the image. Indeed, to obtain the weight vectors of the database classes, we used an offline optimization mechanism based on Differential Evolution (DE) algorithm (Storn & Price, 1995).

1.3.2 Developing effective feature dimensionality reduction methods

In order to reduce the high-dimensional features derived from images, in general and, histopathology images in particular, we first proposed a dimensionality reduction method based on Particle Swarm Optimization (PSO) algorithm (Kennedy & Eberhart, 1995) for the features extracted through handcrafted descriptors. Then, we proposed another method based

on sine cosine algorithm (SCA) (Mirjalili, 2016) for reducing and optimizing the features extracted through deep learning methods.

- ***A dimensionality reduction method based on Particle Swarm Optimization algorithm for histopathology image retrieval (Boukerma et al, 2023)***: In this work, we introduced a selection scheme based on Particle Swarm Optimization algorithm (Kennedy & Eberhart, 1995) in order to reduce the dimension of the features extracted from histopathology images. To extract the features, we used the Improved Local Binary Patterns (ILBP) descriptor (Jin et al, 2004) which was coded in three color spaces. The extracted features from the three-color spaces were then combined to enhance the retrieval performance. To reduce the high dimension of the combined feature vector, we selected the most relevant patterns of this latter while discarding the irrelevant ones. The selection of the subset of relevant patterns was performed using some optimal weights that were generated using PSO algorithm and assigned to all the patterns composing the combined feature vector.

- ***A pattern selection scheme based on Sine Cosine Algorithm for feature dimensionality reduction (Boukerma et al, 2025b)***: As an extension of the previous work, we proposed another optimization technique based on a relatively recent and computationally efficient algorithm, namely sine cosine algorithm (Mirjalili, 2016), to reduce the dimension of other color-texture features, and using an additional histopathology image dataset. Specifically, we extracted the features by applying the LBP and OCLBP descriptors in a multi-color space; then we combined the resulting features to form more discriminative features, namely Combined-LBPs and Combined-OCLBPs. To reduce the high dimension of these combined local patterns, we proposed a pattern selection scheme based on SCA to pick the most relevant patterns among all the patterns constituting the combined features, while also re-weighting the selected patterns to improve the retrieval performance. To assess the performance of the proposed approach, two histopathology image datasets were used, namely KIMIA Path 960 and Kather-5k.

- ***An effective optimization method for reducing deep features dimension for colon histology image retrieval (Boukerma et al, 2024)*:** In order to take advantage of deep learning techniques, that have been successfully used in CBIR to enhance the retrieval performance, and at the same time reduce the high dimension of the deep features

extracted using these techniques, we proposed in this work an optimization approach as a tool for reducing the deep features dimension. In particular, we used transfer learning on the pre-trained ResNet18 (He et al, 2016) and GoogLeNet (Szegedy et al, 2015) Convolutional Neural Network (CNN) models to extract effective feature representations for colon cancer histology images. Then, to further improve the CBIR performance, we combined the deep features extracted through the two above-mentioned CNNs. Subsequently, to refine and optimize the combined deep features, we applied SCA algorithm to generate the optimal weights that were used to select the most relevant bins from the combined deep features and at therefore reducing its size and the retrieval time.

The effectiveness of all the proposed methods was tested on different image datasets, and the obtained results showed their superiority over some existing methods.

1.4 Thesis outline

This chapter has provided an overview of our study along with the motivation and objectives of the thesis, as well as a summary of our contributions. The reminder of the thesis is structured into six chapters.

Chapter 2 gives an overview on content-based image retrieval. More specifically, this chapter provides some basics about the CBIR field, such as visual features and feature extraction techniques, matching measures, performance evaluation techniques, popular datasets in CBIR, etc.

Chapter 3 highlights local pattern descriptors which are widely employed in CBIR for extracting texture features. In particular, a special focus is given to local binary patterns (LBP) descriptor (Ojala et al, 1996) some of its successful extensions.

Chapter 4 reviews the main optimization and learning methods widely used in CBIR. First, we provide a common categorization of the optimization methods, focusing mainly on those commonly used CBIR. More particularly, we emphasize in this part on the metaheuristic methods which constitute a large part of the optimization approaches used in CBIR works. In the second part of this chapter, we investigate learning methods while presenting their taxonomy. In particular, we highlight deep learning methods and specifically conventional neural networks (CNNs) which are widely employed in recent CBIR works.

Chapter 5 is devoted to the first part of our contributions which involve the development of discriminative feature representations for effective image retrieval. The contributions "O-OCLBP" and "O-IOCLBP" descriptors are firstly presented. Then, the "Dynamic pattern weighting" approach is introduced.

Chapter 6 presents the second part of our contributions which include the proposed optimization and learning methods for feature dimensionality reduction. Firstly, we describe the proposed dimensionality reduction method based on particle swarm optimization algorithm for histopathology image retrieval. Secondly, we detail the expanded version of this method which deals with more local pattern methods and more histopathology image datasets. Finally, we present our third method, which leverages deep learning and sine cosine algorithm to achieve an optimized feature space for colon histology image retrieval.

C***hapter 7*** concludes the thesis by highlighting the contributions and outcomes of this research, and outlining the potential directions for future work that can extend this research.

Part II

State of the Art of CBIR

Chapter 2

Content-Based Image Retrieval

2.1 Introduction

The research issues addressed in this thesis are related to content-based image retrieval. To elucidate this area, we introduce in this chapter some fundamental concepts related to CBIR. We first provide a brief overview of retrieval systems, highlighting specifically CBIR systems and their principal components. Then, we outline the feature extraction component along with the most common descriptors used for extracting the features. Further basic concepts related to CBIR such as matching measures, performance evaluation metrics and image datasets are also introduced in this chapter. Finally, we discuss some prominent applications of CBIR.

2.2 A broad view of image retrieval systems

Image retrieval is one of the most successful applications of information retrieval field (Roshdi & Roohparvar, 2015). Basically, information retrieval systems were developed to search and retrieve knowledge-based information from large collections of data (generally documents), whereas image retrieval systems are specifically designed for searching and retrieving images from large image databases (Roshdi & Roohparvar, 2015; Tyagi, 2017). Both information and image retrieval are further described in the following.

2.2.1 Information retrieval

Historically, information retrieval systems referred to text retrieval systems, due to the fact that they were designed to handle textual documents (Chowdhury, 2010). The term "information retrieval" is loosely defined in many contexts. Yet, from the standpoint of computer science, Manning et al. (Manning, 2008) have specifically defined information retrieval as follows: "*Information retrieval is finding material (usually documents) of an*

unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).". The objective of an IR system is then to find relevant information corresponding to a user's query from large collection of data (typically documents). Even though the first information retrieval systems were related to the retrieval of textual documents, the modern ones deal with various types of information such as images, videos, time series and audio (multimedia information) (Chowdhury, 2010; Tyagi, 2017). To deal with the specific nature of the multimedia information, the algorithms and techniques used in information retrieval were largely developed to facilitate the storage, indexing and search of text as well as multimedia information.

As our main key interest in this thesis is focused on searching and exploring digital images, we concentrate in the next section on information retrieval systems that handle this specific type of data (*images*).

2.2.2 Image retrieval systems

Image retrieval systems are considered as an extension to general information retrieval systems, that use the foundational principles of information retrieval to handle still images (Enser, 2008). Indeed, the gigantic growth in the volume and variety of digital images imposed the need to adapt and expand the information retrieval techniques to manage the accumulated digital image databases and to retrieve relevant images from these huge databases.

Image retrieval is defined as searching images from an image database so that the retrieved images are semantically relevant to the user query (Chowdhury, 2010; Tyagi, 2017). The aim is therefore to extract meaningful information from the database images in the sense that the images returned to the user by the system are semantically the closest to its query. To achieve this objective image retrieval relies, in addition to the IR techniques, on different methods and techniques from computer vision, machine learning, multimedia research, and human-computer interaction (Unar et al, 2019). Overall, image retrieval methods can be classified into three groups: text-based image retrieval (TBIR), content-based image retrieval (CBIR) and semantic-based image retrieval (SBIR) (Alkhawlani et al, 2015; Nhi et al, 2020). While the first approach is based on the textual information in images (i.e., textual annotations in form of keywords or tags), the second relies on the content information in images (i.e., visual features such as color, texture and shape). As for the third approach, a

semantic analysis is associated with CBIR (for instance by using the visual content of images with the textual information of images) to search for relevant images (Nhi et al, 2020).

2.2.2.1 Text-Based Image Retrieval

Text-based image retrieval , which dates back to the late 1970s, is one of the early approaches of image retrieval. TBIR is the most widely used retrieval approach, due to its simple process of search based on matching textual queries with manually textual annotations of the database images (Dey et al, 2019; Jan et al, 2020). In fact, many well-known search engines such as Google, Bing and Yahoo Image Search use this approach for searching and retrieving image. Moreover, several TBIR systems have been developed, such as: (Escalante et al, 2008; Krishnan et al, 2021; Li et al, 2011; Singhania & Tripathy, 2021; Unar et al, 2019). In TBIR, images are indexed according to their content using text descriptions in the form of keywords or tags that can represent the image caption, filename, title of the web page, etc. To search images, the user provides a keyword that represents the textual query, then a comparison between the textual feature of the query and those of the database images is performed by the system to find similar images whose surrounding texts are close to the query keyword. In TBIR, the search task therefore relies on the annotation, which can be either manual or automatic (Tyagi, 2017). In traditional TBIR systems, images are manually annotated by humans; however, in the TBIR systems based on the automatic annotation, the images are annotated by the image retrieval system (Tyagi, 2017). Particularly, TBIR based on manual annotation is appropriate when dealing with small image databases; however, it becomes impractical in the case of large image databases (Alkhawlani et al, 2015; Tyagi, 2017). This is because annotating manually each image of the database with the most representative keyword or tag needs a huge number of human labors, and this is an expensive task whether in terms of effort or time. Another limitation of the manual annotation is that it is often a subjective task, and this is due to the fact that the annotators may have different perceptions and comprehensions to the same image (i.e.; the same visual content) (Alkhawlani et al, 2015; Tyagi, 2017). The manual annotation is also context-sensitive, incomplete and language-dependent (Alkhawlani et al, 2015; Tyagi, 2017). However, despite the fact that the automatic image annotation facilitates the indexing and search tasks, the generation of significant texts automatically for a large collection of images is not feasible (Tyagi, 2017).

Because of the above-mentioned limitations of TBIR (whether that based on manual or automatic annotation), the need for other alternatives to extract useful information from

images and retrieve effectively and efficiently relevant images became apparent. In that context, Content-based image retrieval has emerged as an attractive alternative to TBIR by exploiting the visual content of images rather than using simple text-based queries and annotation.

2.2.2.2 Content-Based Image Retrieval

The Concept of CBIR has been explored since the early 1990's. Actually, the nomenclature 'Content-Based Image Retrieval' derives from the fact that this technique analyzes the explicit content of digital images, which indeed represents their pixels (Enser, 2008). More precisely, the term "content" in the context of CBIR refers to the actual content of the image which typically includes: color, texture, shape, edges or any other information that can be extracted from the image itself. So contrary to TBIR systems which rely on the metadata associated with images (such as keywords and tags), CBIR systems analyze the visual contents of images. Query By Image Content (QBIC) and Content-Based Visual Information Retrieval (CBVIR) are also other well- known names for CBIR (Singh & Ahmad, 2014).

Generally speaking, the task of a CBIR system is to retrieve similar images from a large image database given an input image, based on the images contents. In CBIR, when a user inputs a query image, the system extracts the visual features from the query image and store them as a feature vector. Then, the image query features are compared to those of the image database. It is worth noting that the query image features are extracted using the same feature extraction method used for extracting the image database (Tyagi, 2017). The comparison between the query image features and those of the database is carried out by calculating the distance between them using a certain matching measure. The database images are subsequently ranked based on their similarity to the query image, and the most similar ones are then displayed by the system to the user. Actually, CBIR is desirable because searching images based on their visual contents, which are closer to the human perception, is more efficient and effective than using the text annotations and metadata associated with images (Tyagi, 2017). Typically, CBIR systems result in more relevant images compared to the text-based image search engines which provide a lot of garbage in the retrieval result (Singh & Ahmad, 2014; Tyagi, 2017). In addition, CBIR is a fully automated technique that does not require a considerable time for indexing images as in the manual annotation process of TBIR (Singh & Ahmad, 2014; Tyagi, 2017). Hence, the 1990s has marked the lunch of many CBIR systems, such as QBIC (Flickner et al, 1995), Photobook (Pentland et al, 1996),

VisualSEEK (Smith & Chang, 1997), MARS (Rui et al, 1997) and Netra (Ma & Manjunath, 1999). These systems and others, which have been developed subsequently (quoted before in section 1.1), had great appeal due to their potential to represent the perceptual content of images and thereby improving the retrieval of relevant similar images. However, despite their success, these systems have faced a great challenge that has attracted and continues to attract the interest of researchers until nowadays. This challenge known as "semantic gap" (Smeulders et al, 2000) arises essentially from the lack of correlation between the semantic meanings that the user typically associates with images (high level concepts) and the visual features automatically extracted by the CBIR system (low-level features such as color, texture and shape). In other words, it is the difference between what the user means and expects when searching for images and what the machine provides as similar images. Indeed, as the machine interpretation of images in CBIR is based on low-level features, which does not always align with the human perception of images, the commercial CBIR systems launched in the first half of the 1990s did not gain significant market expansion (Eakins, 2002). Indeed, CBIR techniques are highly effective in situations where the text associated with images (in the case of TBIR) cannot capture the perceptual saliency of visual features used in CBIR (Enser, 2008). However, the performance of CBIR systems remains far from user intent due to the differences between the low-level visual features extracted from images and the high-level concepts associated by the user to the same images. Although there are many other issues in CBIR that we will detail in the following sections, the semantic gap remains the main issue. In fact, reducing the semantic gap is a key focus today in developing more effective CBIR systems.

2.2.2.3 Semantic-Based Image Retrieval

To capture semantics description of images, Semantic-Based Image Retrieval (SBIR) has been introduced in 2000s (Wang et al, 2010). In particular, SBIR systems have attempted to bridge the semantic gap by using semantic terms to represent the image content (Wang et al, 2010). Indeed, these semantic terms make it easier for the user to express the meanings that he has in mind for an image when making a search by means of text query which is more intuitive and generally desirable by users (Wang et al, 2010). The overall process of SBIR involves first extracting low-level visual features (object/region features) from images. Then, these object/region features are mapped into semantics to define the semantic content of images (Alkhawlani et al, 2015). The semantic features are then stored in a database and used subsequently to retrieve for related images (Alkhawlani et al, 2015). The semantic mapping

process can be realized using either supervised or unsupervised learning (Alkhawlani et al, 2015). The essential techniques used in SBIR include, semantic templates, image semantic tag assignment, object detection and recognition, image classification and bag-of-visual-words (Pradhan et al, 2018). Among the notable SBIR methods are those proposed in: (Carneiro et al, 2007; Duygulu et al, 2002; Fei-Fei & Perona, 2005; Liu et al, 2008; Mori et al, 1999). Despite the advantages of SBIR which are mainly (1) the capacity of a more thoughtful processing of images, and (2) the use of natural language queries, it nevertheless has several limitations (Ramesh Babu & Reddy, 2018). The main limitation of SBIR systems is the difficulty of describing the semantic content of images (Ramesh Babu & Reddy, 2018). In addition, the semantic terms used to express the meanings of images can often be subjective and have different semantic interpretations (Ramesh Babu & Reddy, 2018). For instance, the term "Freedom" can be depicted by various symbols according to the cultural context, such as open birdcage, flying birds, open hand, broken chains, open road, statue of Liberty, etc. (see Figure 2.1). Also, the effectiveness of an SBIR system is limited and strongly dependent on the size of its vocabulary (Ramesh Babu & Reddy, 2018). Indeed, if the vocabulary doesn't incorporate some specific terms, the SBIR system may fail to retrieve relevant images.

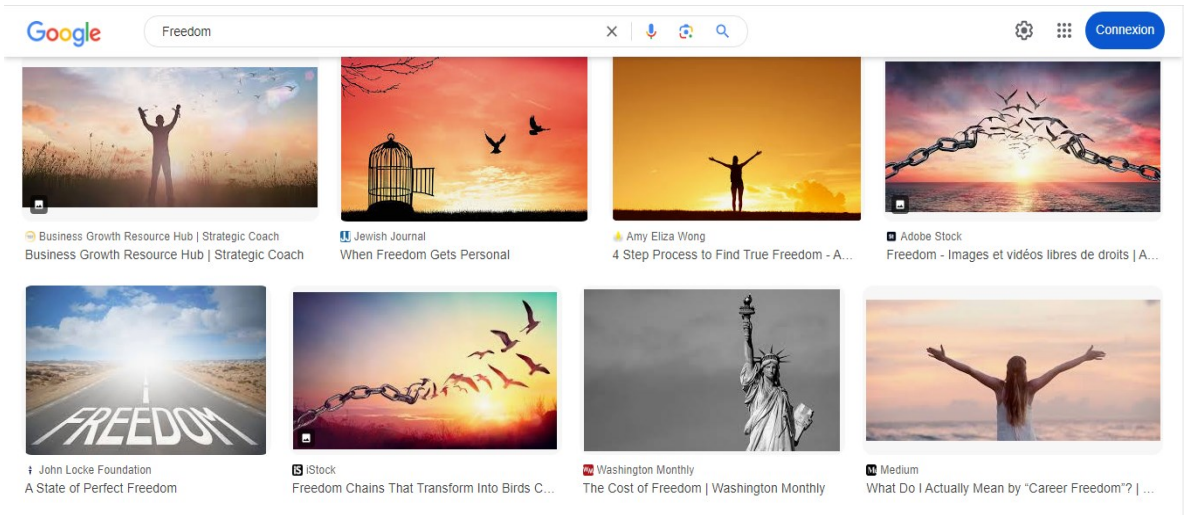


Figure 2. 1: An example of a text query from Google search engine

In light of the above, CBIR systems might be considered more favorable than TBIR and SBIR systems. Lew et al. (Lew et al, 2006) reported that CBIR methods are required when text annotations are missing or incomplete, and may enhance the retrieval performance even if text annotations exist by adding a further overview to the media collections. Compared to SBIR which requires high pre-processing cost and complexity (whether in terms of storage

space or CPU time), the retrieval process in CBIR is more resource-efficient as it is direct (i.e., relies on actual visual content of images) and does not require either the annotations used in TBIR or the additional metadata used in SBIR (Pradhan et al, 2018).

2.3 General framework of a CBIR System

In this section, we present the general framework of a CBIR system along with its principal components.

A standard CBIR system comprises two fundamental modules or mechanisms: *indexing* which is an offline mechanism and *retrieval* (or *search*) which is an online mechanism (Vassilieva, 2009; Vijaykumar, 2020). The indexing and retrieval modules are based on image representation and the used matching measure, respectively (Li et al, 2021b; Vassilieva, 2009). Particularly, feature representation and matching measure are the key components of any CBIR system, and the success of any method or algorithm in the context of CBIR is related to these two components which significantly influence the retrieval performance (Vassilieva, 2009). In addition to the feature representation of the extracted features and the matching measure, the general framework of CBIR involves other important components, namely query image (input), image database, feature database, ranking, result display (output).

As shown in Figure 2.2, all components of CBIR are correlated and each component plays a well-defined role to deliver relevant images to the user. A CBIR system operates by first coding each image in the database and storing the coded images in an indexing database. This phase which performs offline starts by extracting visual features from images (either conventional features extracted using handcrafted descriptors such as color, texture and shape or deep features extracted through deep learning models such as convolutional neural networks); then, the extracted features are converted into a structured format (feature vectors) and stored in the feature database (Li et al, 2021b; Vijaykumar, 2020). In particular, to enhance the quality of images and prepare them for feature extraction, a pre-processing step of the images of the database might precede the extraction of features (Alzu'bi et al, 2015). Depending on the aim of the CBIR application, several preprocessing techniques might be applied to the database images, including segmentation of images, resizing, rescaling, de-noising, etc. (Alzu'bi et al, 2015). The offline phase is of prime importance in the CBIR framework because indexing the extracted features in a structured form allows a fast image

matching and an efficient search during the online phase. The latter starts when a user submits a query image as an input to the system. Then, the feature vector of the query image is extracted using the same feature extraction method adopted in the offline phase for building the index (the feature database) (Li et al, 2021b). Thereafter, a feature matching between the feature vector of the query image and those of all images in the database is carried out using a matching measure. The latter computes the distance between the features to establish how close each image in the database is to the query image. The database images are therefore ranked based on their matching scores, and those with higher scores (i.e., smaller distances) are considered as the most similar images and presented to the user. In some CBIR approaches, the search might be refined by giving the user the possibility to decide which images are relevant to his query, it is what is known as "relevance feedback" (Alzu'bi et al, 2015). In fact, based on the user feedback on the relevance of the returned images, a new search might be performed to return another set of images that matches the user's needs and consequently improve the retrieval performance.

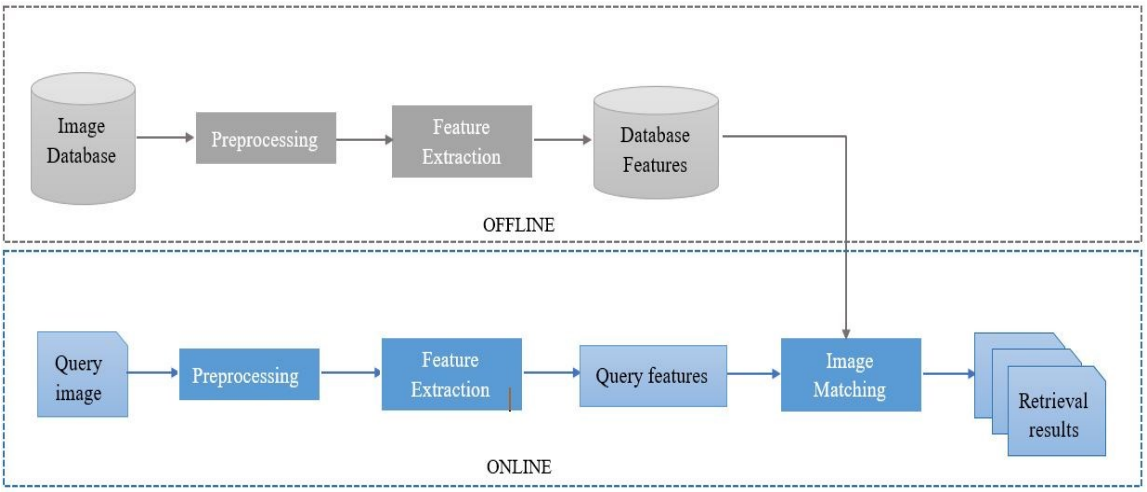


Figure 2. 2: General framework of a Content-based image retrieval system

In the following, we will highlight the main components of CBIR.

2.4 Feature Extraction

Feature extraction is a fundamental process in CBIR and the key to any CBIR technique (Hameed et al, 2021; Khokher & Talwar, 2012). The goal of feature extraction is to convert the visual information that aligns with the human perception (such as color, texture, shape, etc.) into a numerical form that can be handled and analyzed by the machine (Hameed et al, 2021). In general terms, feature extraction can be defined as the process of converting the

input data into a set of features, or what is called features vector, in order to get significant information from the original data and represent this information in a reduced space (Kumar & Bhatia, 2014). In a way, feature extraction can be considered as a sort of dimensionality reduction, because the information extracted from the original data (i.e., the full-size input which may include redundant data) is represented in a lower dimensionality space including only the most relevant data (Kumar & Bhatia, 2014). Typically, two approaches of feature extraction are recognized: *handcrafted feature extraction* and *learning-based feature extraction* (Hameed et al, 2021; Zhou et al, 2017).

2.4.1 Handcrafted feature extraction

In this approach, features are manually designed to capture the visual content of images (Zhou et al, 2017). The features extracted using handcrafted methods are usually in form of low-level features (Hameed et al, 2021; Zhou et al, 2017). In turn, low-level features may be either global or local (Rout et al, 2021). Global features such as color, texture and shape capture the visual information from the entire image. Whereas local features capture the information from some key points or parts of the image such corners, points, blobs or edges. Extracting features from the entire image as a whole makes the global features more efficient for image retrieval as they allow for a fast feature extraction and image matching (Hameed et al, 2021). Local features on the other hand has the advantage of being more robust to various transformations such as scale, rotation, background, translation, etc. (Hameed et al, 2021). Color, texture, shape, and spatial information are the common low-level features in CBIR.

2.4.1.1 Color features

Particularly, color is the most commonly used global feature in CBIR (Hameed et al, 2021; Salmi & Boucheham, 2014). The importance gained by color is due to its effectiveness and its power of discrimination (Haji et al, 2019; Salmi & Boucheham, 2014). The latter is gained due to the fact that the human eye can naturally perceive and distinguish objects and images according to their colors (Haji et al, 2019; Hameed et al, 2021). Actually, color features of an image represent the intensities of the image pixels derived from any color channel of the color image (Nalini & Malleswari, 2016). Thus, color features are computed according to the color channels within a specific color space (Hameed et al, 2021). The frequently used color spaces in CBIR are RGB, HSV, YCbCr, and CIELAB (Hameed et al, 2021).

1) Color spaces

− **RGB:** This color model is commonly used in digital imaging and video devices (Srivastava et al, 2023). It stands for the primary colors “Red”, “Green”, and “Blue”. To form colors, these three primary colors are combined in different intensities. A color is then represented by the three values (R, G, B). In fact, the RGB space can be represented by a cube (see Figure 2.3), where the origin of this cube, defined by the coordinates (0,0,0), represents the black color and the point located at the coordinates (1, 1, 1) represents the white color (Tyagi, 2017). In digital systems, the RGB values range from [0,255]. For instance, the triplet (255, 0, 0) represents the red color, (0, 255, 0) is to form green, (0, 0, 255) for the blue color, (0, 0, 0) represents black, (255, 255, 255) for white and (128, 128, 128) for gray. In real-world image processing tasks, the RGB model is not effective because modifying all the three intensity values to form the desired color is a complicate task (Nalini & Malleswari, 2016). However, this model is more effective in situations where color addition is necessary and wide range of colors is desired (Srivastava et al, 2023). Thus, RGB model is mostly used for displaying images in electronic systems where colors are represented on screens, such as computer monitors, TV screens, digital photography, etc. (Srivastava et al, 2023).

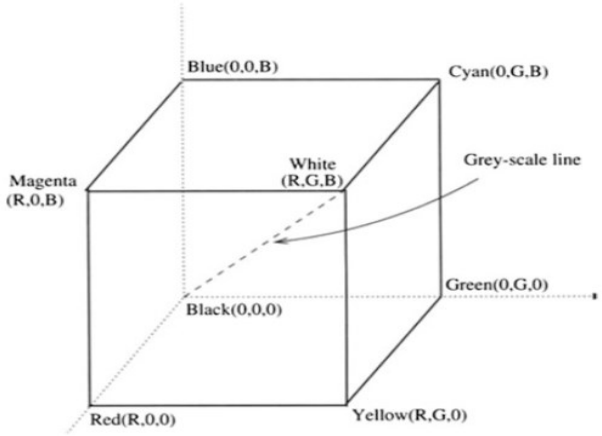


Figure 2. 3: RGB space (Tyagi, 2017)

− **HSV:** Compared to RGB, the hue-saturation-value (HSV) model is closer to human perception (Nalini & Malleswari, 2016). In this model, hue (H) represents the basic color (such as green, yellow, red, etc.) that gives a unique visual impression (Srivastava et al, 2023), and which ranges from 0 to 360°. Saturation (S) signifies the amount or purity of color, which ranges from 0 to 100°. While value (V) represents the brightness of the color, ranging from 0 to 100°. In addition to providing a wide range of colors, HSV model allows for a quantitative manipulation and analysis of color values that can be adjusted

independently (Srivastava et al, 2023). That is why this model is widely employed in image processing (Srivastava et al, 2023). The HSV space is illustrated in Figure 2.4.

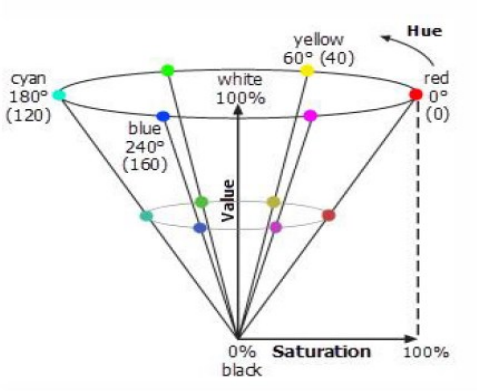


Figure 2. 4:HSV space (Srivastava et al, 2023)

− **YCbCr:** In this color space (depicted in Figure 2.5), luminance and chrominance components are used to represent images. Precisely, Y stands for the luminance component, and ranges from [16,235]; while Cb and Cr represent the chrominance components, ranging from [16,240]. This model is widely employed in digital television transmission (Mane & Bawane, 2014), and successful in skin color classification and detection (Nalini & Malleswari, 2016; Shaik et al, 2015).

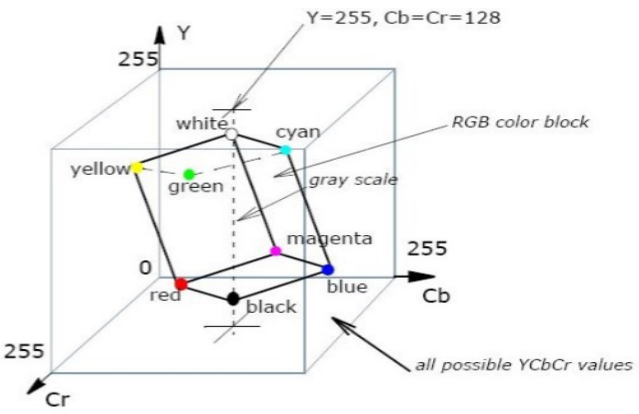


Figure 2. 5: YCbCr space (Khediri et al, 2021)

−**CIELAB:** Like HSV, CIELAB (also known as Lab) color space is also based on human perception (Mane & Bawane, 2014; Nalini & Malleswari, 2016). This model is represented by three components, the first component L* which consists in the brightness; the second component a* represents the color spectrum that varies from green to red; and the third component is b* which denotes the color spectrum varying from blue to yellow. A color is then described by the values of these three components or channels. CIELAB space is illustrated in Figure 2.6.

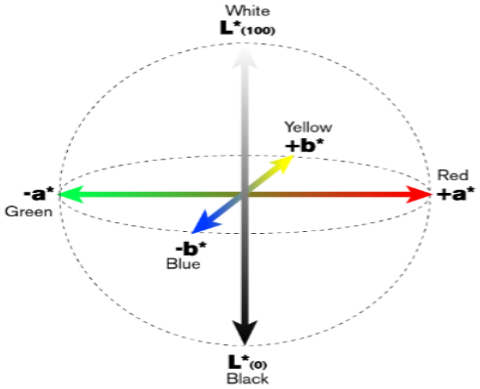
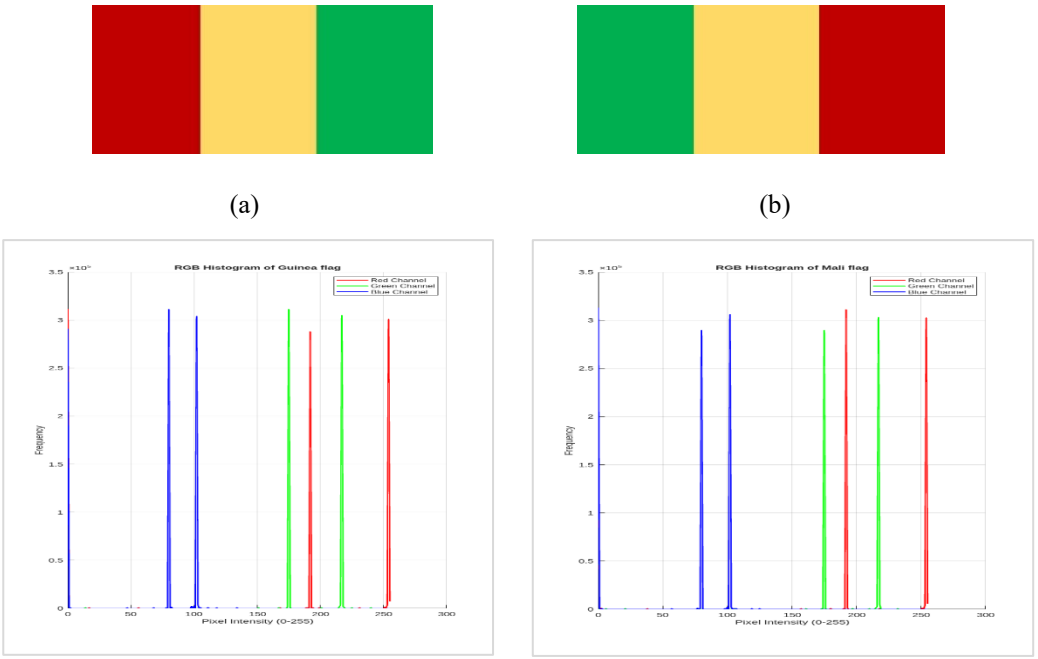


Figure 2. 6: CIELAB space (Khediri et al, 2021)

2) Color descriptors

To characterize these color spaces, many color descriptors were used (Hameed et al, 2021), including color histograms, color correlograms, color moments, dominant color descriptors.

- **Color Histogram:** Color histogram was introduced in 1990 by Swain and Ballard for image matching (Swain & Ballard, 1990). It is a simple yet powerful color descriptor, and commonly used in content based-image retrieval and image processing (Admile et al, 2021; Kelishadrokhi et al, 2023; Kumar & Saravanan, 2013; Nalini & Malleswari, 2016; Narwade & Kumar, 2016). Actually, a color histogram is constructed of three histograms created separately for each color channel (such as red, green, and blue in RGB color space). Nonetheless, the color histogram may also be created in other color spaces (Nalini & Malleswari, 2016). To describe the color distribution of an image, bins are used in color histograms to count the frequency of pixels that correspond to a specific range of color intensities. For example, if a pixel has a blue value of 100, the corresponding bin for the blue channel will be incremented. In addition to their simplicity, invariance to translations and rotations is another strength of color histograms (Srivastava et al, 2023). In fact, they are relatively less affected by changes in perspective, scale, and orientation (Srivastava et al, 2023). Particularly, color histograms are useful in case the spatial arrangement of colors in the image is less important. However, as the spatial arrangement of colors is not taken into consideration in color histograms, two images may have the same histogram (Srivastava et al, 2023). Figure 2.7 illustrates an example of two different images with similar color histograms. Figure 2.7(a) represents an image of the Guinean flag and Figure 2.7(c) depicts its RGB histogram, whereas Figure 2.7(b) represents an image of the Malian flag and Figure 2.7(d) illustrates its RGB histogram.



(a) (b)

(c) (d)

Figure 2. 7: An example of two different images with similar color histograms: (a) Guinean flag, (b) Malian flag, (c) RGB histogram of the Guinean flag, (d) GB histogram of the Malian flag

- **Color correlogram:** This color descriptor was proposed in 1997 by Jing Huang et al. (Huang et al, 1997) for image indexing purpose. Unlike color histograms which consider only the color distribution of colors to characterize images, color correlogram focuses on the spatial correlation of colors in the image. Actually, a color correlogram is structured as a table that is indexed according to pairs of colors (Tyagi, 2017). This table determines the probability of occurrence of a specific color pair (c_i,c_j) at a specific distance k. For example, in RGB color space and for the entry (red, green) at the distance k=2, if the probability of occurrence is 0.5 that means that there is a chance of 50% to find a red pixel distant by 2 pixels from a green pixel. Mathematically, a color correlogram is defined as in equation (2.1) (Park et al, 2007). Compared to color histograms, correlograms are more efficient for image indexing and retrieval, since they capture the spatial relationship between pairs of colors (Nalini & Malleswari, 2016). They are also more robust to image transformations such as zooming, rotation and scaling (Nalini & Malleswari, 2016). However, correlograms usually lead to a high complexity whether in terms of memory space or computation time (Tyagi, 2017). This problem is accentuated in the case of large databases of the fact that the representation of correlograms consists mainly of tables of $O(m^2d)$ complexity (Tyagi, 2017). Where m is the number of images in the database and d is the feature dimensionality. In the

context of CBIR, many works have used color correlograms such as (Park et al, 2007; Somnugpong & Khiewwan, 2016; Soni et al, 2021).

$$\gamma_{c_i,c_j}^{(k)}=Pr_{p_1\in I_{c_i},p_2\in I}\left[p_2 \in I_{c_j}\middle||p_1-p_2|=k\right] \quad (2.1)$$

Here, p_1 and p_2 are two pixels from the image I, and $|p_1-p_2|$ is the distance between these two pixels. I_{c_i} and I_{c_j} denote the colors of the image pixels p_i and p_j, respectively. The k-th entry of the table (which represents the color correlogram) for a color pair (c_i, c_j) indicates the probability that a pixel of color c_j occurs at a distance k from a pixel of color c_i.

– **Color moments:** To overcome limitations of color histograms which does not consider the spatial arrangement of colors in images, Stricker and Orengo (Stricker & Orengo, 1995) proposed a framework using color moments to represent features for image retrieval. In fact, they have proved that features extracted using color moments are more robust than those extracted using color histograms (Shih & Chen, 2002). Color moments are statistical descriptors that determine the probability of colors distribution (Viet Tran, 2003). To compute color moment features, three components are computed for each color channel, including first moment (mean), second moment (variance) and third moment (skewness). The first, second and third moments are computed as in equations (2.2), (2.3) and (2.4), respectively. These moments are then combined in a single descriptor that can represent the color content of images. Therefore, color moment features are formed from nine values (three moments for each color channel), making them more compact compared to other color descriptors but less discriminative (Tyagi, 2017). Thus, color moments are usually used in CBIR as a first step to reduce the search space; then, other discriminant color features may be used to improve the retrieval performance (Tyagi, 2017).

$$\mu_i=\frac{1}{N}\sum_{j=1}^{N}f_{ij} \quad (2.2)$$

$$\sigma_i=\left(\frac{1}{N}\sum_{j=1}^{N}\left(f_{ij}-\mu_i\right)^2\right)^{1/2} \quad (2.3)$$

$$s_i=\left(\frac{1}{N}\sum_{j=1}^{N}\left(f_{ij}-\mu_i\right)^3\right)^{1/3} \quad (2.4)$$

Here, N refers to the total number of the image pixels, and f_{ij} denotes the value of the i-th color component of the pixel j.

Some of the works that have employed color moments in CBIR include (Hassan et al, 2023; Shih & Chen, 2002; Singh & Hemachandran, 2012).

− **Dominant color descriptor:** Contrary to the previous color descriptors which describe the distribution of all colors within an image or the relationship between them, dominant color descriptors (DCD) extract only few colors (the most prevalent) from the image (Nalini & Malleswari, 2016). DCD is an MPEG-7 color descriptor (ISO/MPEG-7, 2001) which identifies the salient color distributions in a specific area or region of interest within an image. DCD is actually based on two primary components, representative colors and their percentages (Nalini & Malleswari, 2016). Mathematically, the dominant color descriptor is represented by as a set of triplets, as in equation (2.5). Each triplet consists of the representative 3D color vector (c_i), the percentage of that color in the image (p_i), and its variance v_i (Tyagi, 2017). To determine the representative colors, techniques such as clustering are used to group the similar colors existing in a specific region/ image (Nalini & Malleswari, 2016). Then, the percentage of each color in the image is calculated (Nalini & Malleswari, 2016). Indeed, the percentage of a specific color is the number of pixels associated to the corresponding color cluster. In other terms, DCD computes the percentage of the image occupied by each dominant color, which facilitates the comparison of images and makes the representation of the image colors more effective, compact and intuitive (Rao et al, 2011a). However, the comparison of images using DCD descriptors is not aligned with the human perception and may lead to incorrect results where the supposed similar images have similar dominant colors but visually appear different (Po & Wong, 2004). That is why DCD has often been combined with other descriptors in many CBIR works, such as in (Rao et al, 2011a; Singh et al, 2017; Xie et al, 2020).

$$\text{DCD}=\{\{(c_i, p_i, v_i): i = 1, \ldots, n\}, s\}, \quad 1 \leq n \leq 8 \tag{2.5}$$

Here, s is the spatial coherence of the dominant colors in the image, n is the total number of color clusters in the image region (usually varies from 1 to 8)) and $p_1+\cdots+p_n=1$ (Tyagi, 2017).

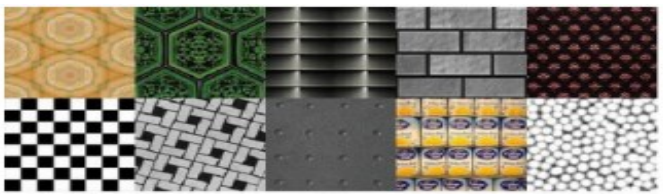
2.4.1.2 Texture features

Texture is an important visual characteristic of image that captures the spatial arrangement of pixels within an image (Srivastava et al, 2023). Indeed, it is difficult to provide a precise definition for texture. The latter has been defined from different perspectives. From an analytical perspective, Tamura et al. (Tamura et al, 1978) regarded texture as "what

constitutes a macroscopic region". Its structure consists of repetitive patterns in which elements or primitives are organized following a "placement rule" (Rosenfeld, 1975). According to Haralick (Haralick, 1979) an image texture is characterized by the number and types of its primitives as well as the spatial arrangement or layout of these tonal primitives. The latter may be spatially arranged in different ways: random organization, pairwise dependence or a dependence of n primitives at the same time. The texture in fact cannot be analyzed without the context of tonal primitives. Sklansky (Sklansky, 1978) defined texture from the segmentation perspective, he considered that "a region in an image has a constant texture if a set of local statistics or other local properties of the picture are constant, slowly varying, or approximately periodic". From another point of view, texture refers to the patterns of the surface of an object within an image (Younus et al, 2015). And since objects are present in the whole natural and real-world images (Haji et al, 2019), texture has gained an increasing importance in many applications such as medical imaging, satellite imagery, agriculture, etc.

In our daily life, we can find two types of textures, natural textured surfaces and artificial textured surfaces (Srivastava et al, 2023). Natural textures or stochastic textures usually occur in nature. This type of textures such as wood, water, stone, grass, and soil exist a lot in real-world images, and they do not follow a specific or regular pattern. Some examples of natural textured surfaces are illustrated in Figure 2.8(a). Artificial textures, however, are more uniform and their texels follow specific pattern. This type of textures such as checks, ceramics, stripes, marble and textured paints, are usually manufactured by humans. Figure 2.8(b) shows some examples of artificial textured surfaces.

(a)



(b)

Figure 2. 8: Example of texture types: (a) natural textured surfaces, (b) artificial textured surfaces (Srivastava et al, 2023)

In CBIR, texture features are widely used and have a significant impact on its performance (Haji et al, 2019). In fact, two images are considered to have similar textures if there is a matching between their spatial pixel arrangements (Srivastava et al, 2023). Although texture is a powerful feature for image retrieval, it has some limitations such as its computation complexity, noise sensitivity and image retrieval accuracy (Hameed et al, 2021; Rout et al, 2021). Usually, texture extraction approaches are categorized into four classes: structural, statistical, model-based and transform (Materka & Strzelecki, 1998).

1) Structural approaches

In general, these approaches perform well with artificial textures exhibiting regular pattern (Tyagi, 2017). They are particularly appropriate for images with large texture primitives or texels (macro-texture), where the shape and basic properties of these primitives are visually dominant and the computation of their placement in the image is more efficient (Srivastava et al, 2023; Tyagi, 2017). Structural methods thus process texture as a set of primitives or texels while analyzing the spatial relationship between these primitives (Tyagi, 2017). The analysis of texture in this approach focus on two aspects: texture element extraction and inference of the placement rule (Srivastava et al, 2023). A texture element (texel), which is a fundamental unit in texture, is a region in the image identified by uniform gray levels. After the identification and extraction of the texture elements, the placement rules allow to determine the positions and placement of these elements in the image. The strength of structural approaches is that they offer a good symbolic representation of the image (Materka & Strzelecki, 1998). However, the extraction of clear texture elements using these approaches may be challenging with the natural textures where their texels are variable and irregular (Materka & Strzelecki, 1998).

2) Statistical approaches

Contrary to the structural methods, statistical approaches perform effectively with textures of small primitives (micro-textures) (Tyagi, 2017). Statistical methods describe the texture by the non-deterministic properties that represent the spatial distribution of the pixel intensities (gray levels) and the relationship between them (Materka & Strzelecki, 1998; Tyagi, 2017). To characterize the statistical properties of the pixel intensities, first-order, second-order, and third-order statistics are used (Srivastava et al, 2023). First-order statistics focus on the intensity values (gray levels) of the pixel individually, without considering the

spatial relationship with the other pixels. Second-order statistics, however, consider the relationships between pairs of pixels. Statistical methods based on second-order statistics have shown better discriminative power than the transform-based and structure-based methods (Materka & Strzelecki, 1998). Among the common second-order statistical descriptors are Gray-Level Co-occurrence Matrix (GLCM) (Haralick et al, 1973), autocorrelation and Local Binary Pattern (LBP) (Ojala et al, 1996). On the other hand, third-order statistics provide more detailed characterization of texture by considering the relationships between triplets of pixels; however, the statistical methods based on third-order statistics, such as third-order moments, are less commonly used because of their high computational complexity (Materka & Strzelecki, 1998).

- **Gray-Level Co-occurrence Matrix:** GLCM is a statistical method which was introduced by Haralick et al. (Haralick et al, 1973) in 1973. Haralick et al. suggested a set of gray-level co-occurrence matrices which determine the spatial relationship between the intensity values of neighboring pixel pairs, based on two parameters: the angular orientation and distance. In other words, gray-level co-occurrence matrix defines how frequently each pair of pixels (i.e., the gray level value of each pixel and its neighbor pixel) appear at a specific distance (d) and angle (θ). Generally, GLCM is calculated according to four main directions or angles (right /0 degrees, down /90 degrees, left /180 degrees, up /270 degrees), with varying distances between neighboring pixels (1 pixel apart, 2 pixels apart, etc.) (Tyagi, 2017). 14 statistical features using gray level co-occurrence matrices were defined by Haralick, including contrast (equation 2.6), entropy (equation 2.7), energy (equation 2.8), homogeneity (equation 2.9), etc. However, only a few of them must be selected in order to decrease the computational cost (Arebey et al, 2011). GLCM is a powerful descriptor for the characterization of texture (Tyagi, 2017). In particular, it has gained an increasing popularity in CBIR due to the fact that the features derived from grayscale images using GLCM align with the human visual perception (Arebey et al, 2011; Nalini & Malleswari, 2016; Srivastava et al, 2020).

$$\text{Contrast} = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} (i-j)^2 P(i,j) \tag{2.6}$$

$$\text{Entropy} = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} P(i,j)\, log P(i,j) \tag{2.7}$$

$$\text{Energy}=\sum_{i=0}^{N_{g-1}}\sum_{j=0}^{N_{g-1}}P(i,j)^2 \tag{2.8}$$

$$\text{Homogeneity}=\sum_{i=0}^{N_{g-1}}\sum_{j=0}^{N_{g-1}}\frac{P(i,j)}{1+(i-j)^2} \tag{2.9}$$

where P(i,j) is a normalized GLCM obtained by dividing all its elements by a defined constant; and N_g is the number of gray levels in the image.

− **Autocorrelation:** This descriptor is mainly used to identify the repetitive patterns in the image texture (Tyagi, 2017). Indeed, the repetitiveness of patterns is a basic characteristic of texture that defines its fineness or coarseness (Tyagi, 2017). To construct the autocorrelation descriptor, the dot product of the original image with shifted copies of this image is calculated. That way, the coarseness of the texture can be detected. In particular, when the patterns are large, the autocorrelation function decreases slowly as the shift (distance) increases (Humeau-Heurtier, 2019). However, when the patterns are small, the autocorrelation function decreases quickly (Humeau-Heurtier, 2019). On the other hand, when the patterns are periodic, the autocorrelation function alternates periodically between decreasing and increasing according to the distance quickly (Humeau-Heurtier, 2019). The autocorrelation p of an image I of size M×N is defined as follows:

$$p(dr,dc)=\frac{\sum_{r=0}^{M}\sum_{c=0}^{N}I(r,c)\,I(r+dr,c+dc)}{\sum_{r=0}^{M}\sum_{c=0}^{N}I^2(r,c)} \tag{2.10}$$

where (r,c) are the coordinates of the image pixel and (dr, dc) are the shifts or displacements (horizontally and vertically).

In fact, the autocorrelation function allows differentiating between regular and random textures by examining the width of the autocorrelation function peak (Tyagi, 2017). In the case of regular textures, we often find peaks and valleys in the autocorrelation function; however, we only find peak at [0, 0], in the case of random textures (Tyagi, 2017). In CBIR, autocorrelation function is usually used to measure the regularity and coarseness of images texture (Dubey et al, 2010; Saravanan & Sathiamoorthy, 2019). Despite its effectiveness in determining repetitive patterns, autocorrelation is not actually a clear way to measure the

texture coarseness (Humeau-Heurtier, 2019). In addition, it is not a precise discriminator of isotropy for natural textures (Humeau-Heurtier, 2019).

- **Local Binary Pattern:** Practically, LBP is the commonly used texture descriptor in image retrieval (Rout et al, 2021). Its popularity is due to its simple implementation and rapid execution (Rout et al, 2021). To construct the basic LBP histogram, LBP codes are computed for each pixel of the image. These codes which represent the local texture (local patterns) of the image are computed for each pixel by comparing its intensity value with those of its surrounding neighbors. Since LBP codes are computed by comparing the intensity values of pixels, LBP descriptor is more invariant to monotonic changes in the grayscale, and consequently more robust in varying lighting conditions (Hameed et al, 2021). To improve the LBP performance, several variants have been proposed for the basic LBP. LBP and its variants have been extensively used and investigated in CBIR research (Arora et al, 2024; Bougueroua & Boucheham, 2014). In particular, LBP method as well as its variants, techniques and approaches based on it; in addition to other concepts related to local pattern methods will be reviewed in depth in Chapter 3.

3) Model-based approaches

In this type of approach, texture is described based on mathematical models such as fractal and stochastic models (Humeau-Heurtier, 2019; Materka & Strzelecki, 1998). To capture the textural characteristics of images, the appropriate type of the model is first selected (fractal, stochastic, random field, etc.); then, the key parameters of the selected model are estimated (Materka & Strzelecki, 1998). In application, the stochastic model has a main drawback relating to the estimation of the model parameters which involve a high computational complexity (Materka & Strzelecki, 1998). The fractal model, on the other hand, has proved to be more discriminative, especially for the characterization of some natural textures; however, it is generally not appropriate for representing local image structures and may miss significant information when the texture has specific orientations (Materka & Strzelecki, 1998). Compared to the stochastic models, some random field models such as autoregressive models are simpler and more efficient in the estimation of parameters. In contrast, Markov random field models are computationally more intensive, and more complex in the estimation of parameters (Humeau-Heurtier, 2019). Nevertheless, Markov random field models have been widely used in CBIR.

−**Markov random field model:** The wide use of this model in CBIR is due to its efficiency in representing the local dependencies between the image pixels, where a pixel intensity depends only on the previous pixel intensity. This is actually derived from the Markov property which implies that the future state of a process depends only on the current state. In this context, an image is modeled as a suite of random variables $X=\{X_s:s\in S\}$, where s is a pixel or region among all the pixels or regions (S) constituting the image. The Markov random field is then defined as in equation (2.11). Among the CBIR works that have used Markov random field models, we mention the work of Backes et al. (Backes et al, 2015), in which Markov random field has been used for segmenting medical images into various textural regions. This texture segmentation along with a fractal analysis have allowed for better distinction of magnetic resonance imaging (MRI) images content, yielding more accurate retrieval results. In another work, Tsai et al. (Tsai et al, 2009) have used Markov random field model of a Gaussian distribution to describe the texture information of all the representative colors in an image. The results of the proposed CBIR system based on the extracted Gaussian Markov random field features have shown robustness to some geometric (such as rotation, translation and distortion), photometric (such as hue, light and contrast) and noise-related transformations.

$$P(X=x)=\frac{e^{-U(x)}}{Z} \tag{2.11}$$

Given that:

$$U(x)=\sum_{c\in C} V_c(x) \tag{2.12}$$

where X is a random field and x is a particular realization of it; Z is the normalization constant; C is a group of neighboring pixels or regions; and $V_c(x)$ is the potential function that indicates the information coming from the neighbors.

4) Transform approaches

Basically, transform approaches differ from the previous approaches in the way of representing images. In this type of approach, an image is represented in a space whose coordinate system provides an interpretation closely linked to the properties of the image texture, like the frequency or the size (Materka & Strzelecki, 1998). The images are thus analyzed in the frequency domain which is specifically appropriate for texture analysis due to the fact that it reveals the texture periodicity and its repetitive patterns (Srivastava et al,

2023). Fourier transform, Gabor transform, and wavelet transform are the most common transform methods used for describing texture (Abdesselam, 2009).

- **Fourier transform:** This method analyzes images in the frequency domain which allows describing the periodic patterns in the image and identify the dominant orientations of these patterns as well as their distribution in the frequency space (Abdesselam, 2009). This enables to provide significant information about the texture directionality and coarseness which are essential in the texture analysis. However, Fourier transform method underperforms in practice because its insufficiency for the spatial localization size (Abdesselam, 2009; Materka & Strzelecki, 1998). In the context of CBIR, many works for texture retrieval were based on this method (Abdesselam, 2009; Bama & Raju, 2010). For instance, in (Bama & Raju, 2010), Fourier transform has been used to extract rotation invariant texture features from photometric texture images in a three-dimensional co-ordinate system. In order to improve the retrieval performance, the proposed method has analyzed the characteristics of rotated and unrotated textures, and farther the Fourier expansion has been applied to achieve a rotation invariance.

- **Gabor Transform:** This method is intended to be used in the spatial domain for the local analysis of texture. Gabor transform exploits Gabor filters for analyzing texture by applying them at local regions in the image and in specific directions (Srivastava et al, 2023). For an image I at the pixel coordinates (x,y), the 2D Gabor filter g(x,y) is defined as in the following equation:

$$g(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right].\exp(2\pi jWx) \tag{2.13}$$

where σ_x and σ_y denote the standard deviation of the Gaussian along the x and y directions, respectively; W is the spatial frequency of the sinusoid in the filter (high W reflects fine textures and low W reflects coarse textures); and j is the imaginary unit ($j=\sqrt{-1}$).

Compared to Fourier transform, Gabor transform methods are more effective for the spatial localization (Materka & Strzelecki, 1998).

In addition, Gabor transforms are appropriate for stationary textures as they are effective in capturing the spatial correlation between the image pixels (Srivastava et al, 2023). However, in practice, natural textures are usually non-stationary and one single filter resolution cannot

determine the localization of a spatial structure (Materka & Strzelecki, 1998). Gabor filters have been widely used in CBIR (Andrysiak & Choraś, 2005; Nemade & Sonavane, 2020; Sastry et al, 2007). For instance, in (Nemade & Sonavane, 2020), a feature extraction algorithm based on Gabor filter has been proposed. The authors have exploited the multi-resolution capability of Gabor filter to extract relevant features. The experimental results of their work have shown that Gabor filter could generate effective image description with small feature size.

- **Wavelet Transform:** Unlike Gabor transform, Wavelet transform performs well for stationary textures as well as for non-stationary textures (Srivastava et al, 2023). It decomposes an image into four frequency components or sub-bands (LL, LH, HL and HH) at various scales. The Wavelet decomposition results in four components of the image which are the approximate, vertical, horizontal and diagonal components of an image. Wavelets have different types such as Haar, Symlet, Daubechies etc. The strength of Walevet transforms is that it can capture the texture information at the appropriate scale due to the varying spatial resolution (Materka & Strzelecki, 1998). In addition, as there are various types of wavelets with different functions, this allows to choose the most suitable wavelet for a specific application (Materka & Strzelecki, 1998). However, the limitation of Wavelet transform is that it is invariant to translations (Materka & Strzelecki, 1998). Medical image retrieval is among the applications that have widely used Wavelet transform (Bhowmick et al, 2020; Ganasala & Prasad, 2020). For instance, in (Bhowmick et al, 2020), a multiresolution analysis using three level Haar wavelet image components has been used for detecting benign and malignant breast lesions in mammograms.

2.4.1.3 Shape features

Shape features describe images based on the geometric characteristics of objects within an image. Usually, these features are extracted after segmenting images into regions or objects. Since achieving robust and accurate segmentation of images is challenging, shape features have a limited use in CBIR for some special applications where objects or regions can be easily identified (Tyagi, 2017). Shape-based image retrieval methods must effectively handle transformations like rotation, scaling and translation, etc. (Nalini & Malleswari, 2016). Shape-based feature methods can be categorized into two categories, boundary-based and region-based methods (Tyagi, 2017). Figure 2.9 depicts boundary-based and region-based representations.

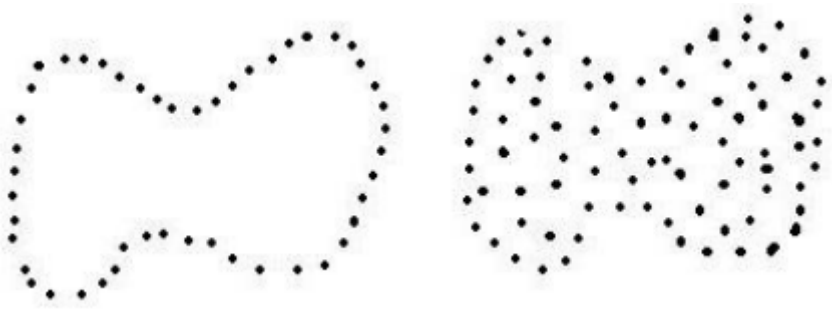


Figure 2. 9: Boundary-based and region-based representations (Khokher & Talwar, 2012)

1) Boundary-based shape features

Features in the boundary-based feature methods are derived from the outside boundary of a region within an image. Among these methods, there are: chain codes, Fourier descriptors, rectilinear shapes, polygonal approximation, etc. One of the most successful boundary-based shape descriptors is Fourier descriptor (Alkhawlani et al, 2015). Fourier descriptor analyzes and describes the shape of a two-dimensional object or contour using Fourier transform of its boundary. By treating the contour of a two-dimensional object as a bounded sequence of contiguous boundary pixels, three forms of contour representations can be defined, including curvature, centroid distance, and complex coordinate function (Tyagi, 2017). The success of Fourier descriptor is due to its low computational complexity, its robustness and its easy normalization (Nalini & Malleswari, 2016). Among the CBIR works that have used Fourier descriptor, we mention the works of Kunttu et al. (Kunttu et al, 2004) and Behnam & Pourghassem (Behnam & Pourghassem, 2013).

2) Region-based shape features

In contrast to the boundary-based approach where the shape features are extracted from the boundary of the region, the region-based approach relies on the features extracted from the entire shape region (Hameed et al, 2021). Region-based approach therefore focuses on the internal characteristics of a region, so that the features are derived from the pixels contained in the region. Moment descriptors are widely used to derive region-based features (Tyagi, 2017). Among these moments, there are: moment invariants, Zernike moments, radial Chebyshev moments, etc. In particular, moment invariants are the most common representative descriptor for the region-based approach (Alkhawlani et al, 2015). Moment invariants are moment functions that are invariant to some specific changes, such as translation, rotation, and scale. To derive shape features using moment invariants, each object

within an image is described by a 14-dimensional vector comprising two groups of normalized moment invariants (Tyagi, 2017). One group is for capturing the outer boundary (the contour) characteristics of the object, and the other is derived from its silhouette (i.e. the whole solid area of the object). Among the CBIR works that have used moment invariants, we mention the works of Park & Kim (Park & Kim, 2005) and Karakasis et al. (Karakasis et al, 2015) .

2.4.1.4 Spatial information features

The limitation of the majority of the previously discussed low-level feature methods (i.e. color, texture and shape) is the lack of spatial information (Hameed et al, 2021). To further clarify this lack, let us assume that we desire to retrieve images of a tree from a large database, using color histogram and texture low-level features. The CBIR system might return images like a green grass field or wooden furniture, because a tree and a green grass field may have similar color histograms (lots of green). Likewise, wooden furniture and a tree may have similar texture features (brown textures). However, the spatial information can help to identify the spatial arrangement of the tree (for example the brown trunk is obviously at the bottom and the green leaves are always at the top). That is why the spatial information extracted from specific regions or objects within an image, including the location of these regions (objects) and the spatial relationship between them, is extremely helpful for retrieving images (Tyagi, 2017). The main limitation of spatial information methods is their high computational complexity (Hameed et al, 2021).

2D strings (Chang et al, 1987) is the most popular representation of spatial information (Tyagi, 2017). In this method, images are projected along the x and y-axis, so that two sets of symbols (strings) are determined. The symbols of the first set (set V) represent the different objects within an image, while the second set (set A) is composed of the symbols that represent the different types of the spatial relationship between the identified objects. Other methods for capturing the spatial information features include 2D G-string (Chang et al, 1989), spatial quad-tree (Samet, 1984), symbolic image (Gudivada & Raghavan, 1995), etc.

In the context of CBIR, using spatial information to represent and search images remains a challenging task (Tyagi, 2017). This issue is due to the difficulties in acquiring a reliable segmentation of images, except in few specific applications (Tyagi, 2017). Yet, some

works such as (Luft & Schiewe, 2021; Ouni et al, 2022) have attempted to overcome this issue by applying certain segmentation frameworks to enhance CBIR performance.

2.4.2 Learning-based feature extraction

In learning-based feature extraction approach, the features are learned automatically from data (Zhou et al, 2017). In fact, low-level features alone are generally not sufficient to describe the complex and abstract content of images (Min & Shuangyuan, 2010). Therefore, high-level features are required to represent image semantics. In this context, several methods and algorithms such as ontologies and machine learning algorithms have been used to derive the semantic features of images (Zhou et al, 2017). The most basic method is to use text to describe the image semantics (Min & Shuangyuan, 2010). In this context, dictionaries (ontologies) and vocabularies have been used to ensure a better interpretation of images based on meaningful words that link to related semantic concepts (Min & Shuangyuan, 2010). In addition to its simplicity, this approach can be helpful to describe some abstract concepts which may be difficult to capture using low-level features alone; however, it is not effective in describing the complex relationship between concepts within an image (Min & Shuangyuan, 2010). For example, describing an image that represents "a cat sitting beside a house" requires more context than just using the words "cat" and "house". To effectively represent high-level concepts of images, other approaches have been used. These approaches can be classified into three classes (Min & Shuangyuan, 2010), including object-ontology methods, relevance feedback methods and machine learning methods (i.e., supervised or unsupervised learning).

Ontologies are used in CBIR to formalize the relationships between the different concepts that an image represents, and therefore describe the implicit semantics of the objects within an image. Ontology was basically employed in text retrieval (Min & Shuangyuan, 2010); then, it has been exploited in image retrieval in order to bridge the semantic gap issue. Object-ontology is a simple vocabulary that consists of various intermediate-level descriptors (Liu et al, 2007). In fact, intermediate-level descriptors are higher-level features which provide a meaningful description of the image content that aligns with human understanding. Relevance Feedback (RF) was applied on image retrieval at the end of the 2000s (Min & Shuangyuan, 2010). It is an interactive process that aims to reduce the semantic gap and improve the retrieval results by incorporating user feedback to the CBIR system. The user then provides the system with additional information which actually consists of a judgment

of certain displayed images (Mosbah & Boucheham, 2014). Briefly, the process of RF starts with an initial retrieval in which the system provides the first returned images based on the user query image (user's input); then, the user selects a subset of images from the currently displayed images to indicate which images are relevant or irrelevant to his query. Based on the user's judgment (feedback), the CBIR system refines the search of images and re-rank the results. This process can be repeated until the user achieves desired results (Min & Shuangyuan, 2010).

Machine learning techniques, including supervised and unsupervised techniques, have been used as formal tools to extract high-level semantic features from images. In CBIR, supervised learning techniques such as image classification are usually used as a preprocessing step for improving the image retrieval efficiency and effectiveness (i.e., fast and accurate retrieval) in large image datasets, or for automatic annotation of images (Min & Shuangyuan, 2010). Support vector machine (SVM) (Cortes & Vapnik, 1995), decision trees and neural networks are frequently employed in CBIR to learn high-level semantics from low-level features (Liu et al, 2007). The general principle of these methods is to train a set of pre-labeled images so that a trained model is constructed with a view to learning new unlabeled images and labeling them with suitable semantic labels. Further details regarding these methods will be provided in chapter 4 (section 4.4.2). In contrast to the supervised learning techniques, the models in unsupervised learning are trained on unlabeled data. Image clustering is a typical unsupervised learning technique (Min & Shuangyuan, 2010), in which similar images are grouped into classes without prior knowledge of their labels. In that context, K-Means is a popular and simple method used for clustering, it was exploited in CBIR to improve its efficiency and effectiveness by grouping either images or regions that have similar characteristics (Liu et al, 2007). Unsupervised learning methods will be further discussed in chapter 4 (section 4.4.1).

2.5 Matching Measures

As mentioned above, feature extraction process plays a crucial role in any CBIR system. Just as importantly, the distance measure used for matching images in CBIR directly impacts the effectiveness of the system (Rout et al, 2021). In fact, the choice of the appropriate distance measure leads to better retrieval results. In the context of CBIR, a distance measure quantifies how the inputted query image is similar (or different) to the images of the database. As

discussed above, images in CBIR are represented by feature vectors. Thus, the query image submitted by the user and all images in the database are mathematically represented by n-dimensional vectors, between which distances are calculated. In the following, we outline the most popular distance metrics used in CBIR.

2.5.1 Euclidean distance

This matching measure, also called *L2* distance, is one of the most used distance metrics in image retrieval (Tyagi, 2017). Mathematically, Euclidean distance is the straight-line distance between two points from a multi-dimensional space. From the perspective of feature vectors, this distance is calculated by comparing the corresponding elements of two feature vectors. The Euclidean distance between two vectors u and v, both of dimension n, is given by equation (2.14); knowing that u=(x_1, x_2, ….,x_n), v =(y_1,y_2,…,y_n) and i=1...n.

$$d_{EU}(u,v)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \tag{2.14}$$

2.5.2 Manhattan distance

Also known as *city-block* or *L1* distance. This matching measure, as compared to the Euclidean distance is more desirable for high-dimensional data and more invariant to translation or reflection along the coordinate axes (Tyagi, 2017). However, it is sensitive to changes in rotation of the coordinate system (Tyagi, 2017). Manhattan distance is also a popular distance metric and commonly used in CBIR, especially, in approaches based on color indexing (Tyagi, 2017). Mathematically, the Manhattan distance between two vectors u and v of dimension n (u = (x_1, x_2, ….,x_n), v = (y_1,y_2,…,y_n) and i=1...n) is the sum of absolute differences of their corresponding elements. Thus, Manhattan distance is defined as follows:

$$d_{MH}(u,v)=\sum_{i=1}^{n}|x_i-y_i| \tag{2.15}$$

2.5.3 Chebyshev distance

It also called chessboard, maximum value metric or L∞ metric. Chebyshev metric quantifies the distance between two vectors by taking the maximum difference among the differences of their corresponding elements. In the case of two-dimensional space, the concept of Chebyshev or Chessboard metric reflects the minimum number of moves or steps that a king in chess could take to transit between each pair of squares on the chessboard (Tyagi, 2017).

The Chebyshev distance between two vectors u and v of dimension n is computed as in equation (2.16); knowing that u=(x_1, x_2, ….,x_n) and v =(y_1,y_2,…,y_n).

$$d_{CH}(u,v)=\max\sqrt{(x_i-y_i)^2} \quad ,i=1\ldots n \tag{2.16}$$

It is worth noting that Euclidean, Manhattan and Chebyshev distances are special cases of *Minkowski distance,* also known as *LP norm* (see equation (2.17)). Actually, Manhattan distance is just Minkowski distance when the order p=1; hence its common name "L1" distance. Similarly, Euclidean distance is Minkowski distance when p=2; hence its name "L2" distance. As for Chebyshev distance, it is also an instance of Minkowski when p tends to infinity; where from its naming "L∞" distance.

$$d_{MK}(u,v)=(\sum_{i=1}^{n}|x_i-y_i|^P)^{1/P} \tag{2.17}$$

2.5.4 Canberra distance

This distance metric is considered as a weighted version of the Manhattan distance (Tyagi, 2017). It is used to measure the similarity (i.e., closeness) of a pair of points in a vector space. Canberra distance is specifically practical when signs indicate differences in kind or type rather than in degree (Tyagi, 2017). In other words, it is useful when signs of values might represent different kinds or categories. For example, in the case of classification, the positive sign might represent the class "Healthy" whereas the negative sign represents the class "Sick". Canberra distance is however highly sensitive to values near zero, leading to biased and less accurate results (Tyagi, 2017). For two vectors u and v of dimension n (u = (x_1, x_2, ….,x_n), v = (y_1,y_2,…,y_n) and i=1...n), Canberra distance is calculated as follows:

$$d_{CA}(u,v)=\sum_{i=1}^{n}\frac{|x_i-y_i|}{|x_i|+|y_i|} \tag{2.18}$$

2.5.5 Square chord distance

It is used to measure the distance between two vectors (or points) by computing the sum of the squared differences between their corresponding elements. Specifically, Square chord distance is simply the Euclidean distance squared, what makes it computationally more efficient than the Euclidean distance. In other words, the calculation of Square chord distance is faster than the Euclidean distance of the fact that Square chord metric does not require the

computation of the square root as in the Euclidean distance. That is why Square chord metric is perfectly suitable for image database applications (Ayyachamy & Manivannan, 2013). Mathematically, Square Chord distance is calculated as follows:

$$d_{SC}(u,v)=\sum_{i=1}^{n}(\sqrt{x_i}-\sqrt{y_i})^2 \tag{2.19}$$

2.5.6 D1 distance

The D1 distance which is also called weighted L1 distance (Wei & Liu, 2020), is a weighted and normalized version of L1 or Manhattan distance. In fact, the normalization factor ($1+x_i+y_i$) allows to reduce the effect of large values (see equation (2.20)). This is useful and very effective in CBIR, where feature values can vary widely in range or scale due to various factors such as brightness, contrast, lighting conditions, etc. Hence, D1 distance is being preferred in CBIR over other matching measures as it could achieve high retrieval performance (Agarwal, 2023; Bougueroua & Boucheham, 2015; 2018). For two vectors u and v of dimension n, the D1 distance is calculated as follows:

$$d_{D1}(u,v)=\sum_{i=1}^{n}\left|\frac{x_i-y_i}{1+x_i+y_i}\right| \tag{2.20}$$

2.6 Performance evaluation

The evaluation of a CBIR system performance is an essential task, as it allows to determine the effectiveness and efficiency of the system and helps researchers in comparing their results to other systems or using different benchmarks. Usually, to quantitatively measure the performance of any CBIR system, various predefined system formulas are used. Evaluating CBIR systems performance using predefined and objective formulas is solicited because the human evaluation is subjective, time-consuming and can be incorrect (Hameed et al, 2021). to provide consistent and reliable evaluations. To achieve consistent and reliable evaluation of CBIR performance, various metrics are used. The choice of the most appropriate evaluation metric depends on various factors such as the used method or approach, the algorithm characteristics and specific context or domain of the problem (Hameed et al, 2021). Some common performance evaluation metrics used in CBIR are discussed below.

2.6.1 Precision

The precision of a CBIR system is defined as the number of the relevant images retrieved (i.e., similar images) among the total number of the retrieved images. P is calculated as follows:

$$P = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ images\ retrieved} \tag{2.21}$$

Generally speaking, in the context of information retrieval, the precision evaluate the efficiency of the search (Hull, 1993). In other words, it determines how successfully the system ignores irrelevant results (irrelevant images in the case of image retrieval). Thus, a high precision value, in the case of image retrieval, indicates that the most of the retrieved images are relevant. For a single query, the precision values range from 0 to 1. To provide a general and comprehensive evaluation of the system, the concept of average precision become more useful (Hull, 1993). Indeed, for a large database, it is more meaningful to compute the average precision across all queries rather than judging the performance of the system based on a limited view (i.e., the precision of just one query). In this context, the Average Precision Rate (APR) is widely used in CBIR to measure its performance (Dubey et al, 2016; Issaoui et al, 2024; Liu et al, 2017b; Pathak & Raju, 2021).

Average Precision Rate: For a specific image dataset, the APR averages the precision values across all query images in the dataset. The formula of APR is given by equation (2.22). Knowing that P(n) is the precision of a query image n, and N is the total number of images in the dataset.

$$APR = \frac{1}{N}\left(\sum_{n=1}^{N} P(n)\right) \tag{2.22}$$

2.6.2 Recall

The recall (R) of a CBIR system is defined as the number of the relevant images retrieved (i.e., similar images) among the total number of relevant images in the dataset. R is calculated as follows:

$$R = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ relevant\ images\ in\ the\ database} \tag{2.23}$$

While the precision determines the efficiency of the search, the recall determines the breadth of the search (Hull, 1993). In other words, the recall measures the completeness of the search, or how many of the relevant results the system has been able to retrieve from all relevant results in the dataset. Like the precision, the recall values range from 0 to 1. In fact, a high recall for a given query refers to the ability of the system to retrieve a significant portion of all relevant images in the database. To obtain more robust and meaningful evaluation of the overall system, calculating the average recall across all queries in the image database is more useful than evaluating and judging the performance of the system on the basis of individual queries (Hull, 1993). The Average Recall Rate (ARR) is commonly used in CBIR to provide a comprehensive view of the overall recall performance (Dubey et al, 2016; Issaoui et al, 2024; Liu et al, 2017b; Pathak & Raju, 2021).

Average Recall Rate: For a given image dataset of N images, the ARR is a performance metric that averages the recall values of all query images in the dataset. Given that R(n) is the recall of the specific query image n, ARR is calculated as follows:

$$\mathrm{ARR} = \frac{1}{\mathrm{N}}\left(\sum_{\mathrm{n}=1}^{\mathrm{N}} \mathrm{R(n)}\right) \tag{2.24}$$

Usually, the average recall rate is used alongside the average precision rate to evaluate the performance of CBIR and understand how successfully the system is balancing the retrieval of relevant images (recall) while ignoring irrelevant ones (precision). Broadly speaking, the APR provides an overall view of the quality of the retrieved results while the ARR determine the quantity of the relevant results that were retrieved.

2.6.3 F-Score

F-Score (or F-measure) is a performance metric that combines both precision and recall into a single measure value, giving a balance between the two. In particular, when the precision and recall measures are close, F-Score consists approximately of their average (Tyagi, 2017).

More generally, the F-score measure represents the harmonic mean, which corresponds, in the case of two numbers, to the division of the square of the geometric mean by their arithmetic mean (Tyagi, 2017). The formula of F-Score is given by the following equation:

$$F - score(n) = 2 \times \frac{Precision(n) \times Recall(n)}{Precision(n) + Recall(n)} \quad (2.25)$$

2.7 Benchmark datasets

In this section, we describe some publicly available image datasets that are frequently used in CBIR. These image datasets can vary from domain-specific (or restricted) datasets to general-purpose datasets (Belattar & Mostefai, 2013; Mezaris et al, 2004), depending on the type of images being analyzed or the application area of these datasets.

2.7.1 General-purpose image datasets

These image datasets contain heterogeneous images from diverse categories. Several general-purpose image datasets are frequently employed in CBIR, such as Corel, Flickr, etc.

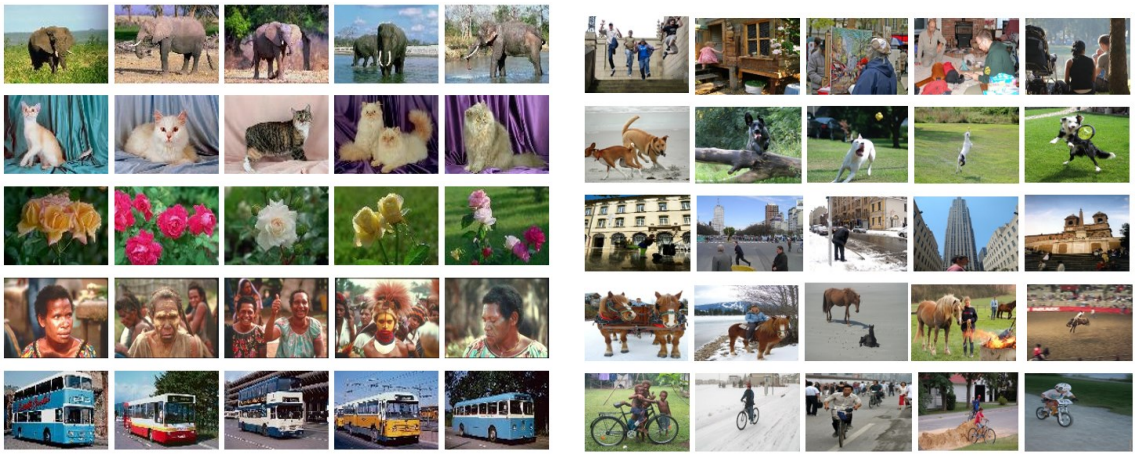
- **Corel:** This image dataset is extensively used in CBIR studies**.** In a survey conducted by Y. Liu et al. (Liu et al, 2007), it was revealed that more than half of CBIR systems in the surveyed papers have employed a subset of Corel image database to evaluate the retrieval performance. The popularity of Corel is due to the fact that it contains diverse images from diverse categories. In fact, Corel dataset includes a large number of color images organized in various classes covering various semantic categories, such as animals, sports, natural scenes, people, buildings, etc. *Corel-1k or Wang* (Wang et al, 2001) , *Corel-5k* (Duygulu et al, 2002) and *Corel-10k* (Li & Wang, 2003) are different subsets with different sizes of Corel dataset that are usually used by researchers in CBIR to meet the specific needs of their studies. In particular, Corel-1K dataset contains 1000 images of 384×256 or 256×384 pixels, divided into 10 classes. Corel-5K dataset includes 5000 images of 187x126 or 126x187 pixels organized in 50 categories. Corel-10K dataset contains 10,000 images of 187x126 or 126x187 pixels organized in 100 categories.

- **MIRFlickr:** The images of MIRFLICKR dataset (Huiskes et al, 2010) are sourced from the social photography platform Flickr. Indeed, this dataset consists of 1 million images randomly downloaded from Flickr website[1], and resized to 500x500

[1] http://flickr.com

pixels. These images are of diverse contents covering a wide range of categories such as landscapes, natural scenes, people, objects, events, etc. Furthermore, MIRFlickr images include the original user tags and metadata. MIRFlickr-25K is a subset of MIRFlickr that contains 25,000 images, usually used for image retrieval tasks as it provides a large variety of images with user tags and metadata.

Sample images from the above-mentioned general-purpose datasets are illustrated in Figure 2.10.



(a) (b)

Figure 2. 10: Sample images from general-purpose image datasets: (a) Corel dataset, (b) Flickr dataset

2.7.2 Domain-specific image datasets

Domain-specific image datasets involve specialized images which are generally dedicated to specific types of applications. Included among these image datasets are collections of texture images such as Brodatz (Brodatz, 1966), KTH-TIPS (Hayman et al, 2004), STex (Kwitt, 2021), USPTex (Backes et al, 2012) and BarkTex (Lakmann, 1998); object images such as Columbia Object Image Library (COIL) (Nene et al, 1996); and medical image datasets such as OASIS-MRI (Marcus et al, 2010), KIMIA Path960 (Kumar et al, 2017) and Kather-5K (Kather et al, 2016) histopathological image datasets.

1) Texture image datasets

- **Brodatz**: It is a widely used texture image dataset which was introduced by Phil Brodatz in 1966 (Brodatz, 1966). Brodatz dataset consists of 112 grayscale texture images with a resolution of 640×640 pixels. Each image in the dataset represents a

different natural or artificial texture, such as grass, sand, bark, leaves, wood, fabrics, paper, leather, brick, etc.

- **KTH-TIPS**: It is a texture image dataset that consists of 10 texture classes, each comprising 81 color images of a resolution of 200 x 200 pixels. KTH-TIPS texture classes represent different kind of material such as aluminum foil, cotton, bread, sponge, cracker, linen, sandpaper, styrofoam, etc. Texture images in this dataset are captured under varying illumination, pose, and scale (9 ratios, 3 poses and 3 illumination conditions). *KTH-TIPS2a* and *KTH-TIPS2b* are extended versions of KTH-TIPS with more images, classes and variability. For instance, KTH-TIPS2b (Caputo et al, 2005) is an extension of KTH-TIPS that has11texture classes, each containing 432 images, and with one additional illumination condition.

- **STex**: The Salzburg Texture image dataset or STex dataset is a color texture image dataset that contains 476 textures, representing different objects and materials such as bark, marble, walls, stones, floor, wood, etc. The images of this dataset are of a resolution of 512x512 pixels. The original STex image dataset has also different versions or subsets, such *STex-512-splitted*[2] dataset which was obtained by splitting the images of STex dataset into 16 non-overlapping tiles, resulting in a dataset of 7616 images with a resolution of 128×128 pixels.

- **USPTex**: This image dataset was created at the University of São Paulo (USP) in Brazil (Backes et al, 2012). It contains 2292 color texture images of 128 × 128 pixels, grouped into 191 classes. These latter represent various natural textures, frequently found in our daily life such as rice, vegetation, cloud, walls, soil, food, fabric, gravel, tiles, etc. USPTex images were captured in the wild, hence under uncontrolled illumination and display conditions.

- **BarkTex**: This dataset which was introduced by Lakmann (Lakmann, 1998) contains 408 color texture images from six different bark texture classes. Each class represents a different tree bark specie captured under natural illumination conditions. *NewBarkTex* is a derived version of BarkTex that was proposed by Porebski et al. (Porebski et al, 2014). It consists of 1632 images with a resolution of 64×64 pixels,

[2] https://www.wavelab.at/sources/STex

which also belong to six tree bark textures. NewBarkTex images were derived by cropping, from the center of images of BarkTex, a region of interest (ROI) of size 128x128 pixels. The latter, was then separated in four sub-images of 64x64 pixels.

Sample images from all the above-mentioned texture image datasets are depicted in Figure 2.11.

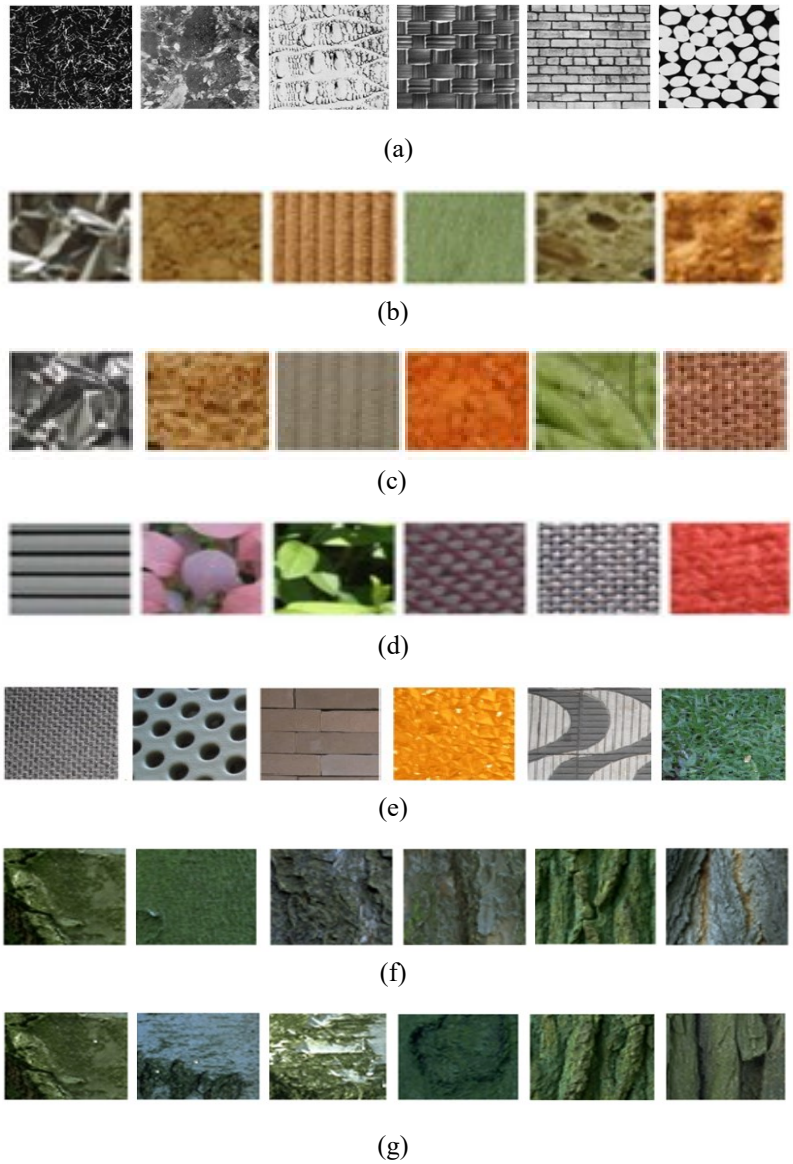


Figure 2. 11: Sample images from texture datasets: (a) Brodatz, (b) KTH-TIPS, (c) KTH-TIPS2b, (d) STex, (e) USPTex, (f) BarkTex, (g) NewBarkTex

2) Object image datasets

- **COIL**: The original version of this dataset, which is *COIL-20* (Columbia Object Image Library) (Nene et al, 1996), is an object-based image dataset of grayscale images that represent different objects from 20 categories such as rubber ducks, toy

cars, and jars. COIL-20 contains 1440 images of size 128×128 pixels taken from varying angles. *COIL-100* (S. A. Nene, 1996) is an extended version of COIL-20 that includes 7200 color images of 100 objects, so that each object has 72 different views (orientations). In fact, COIL-100 has 100 classes, each includes 72 small images in PNG format with a resolution of 128x128 pixels

- **Caltech**: The original Caltech image dataset, i.e., *Caltech-101* (Fei-Fei et al, 2004) was collected from Google images, in a way that it contains different object categories (Griffin et al, 2007). In fact, Caltech-101 contains 9146 images from 101 object categories, plus one additional background category. Each category includes from 40 to 800 images of a resolution of 300× 200 pixels, representing different object classes such as people, animals, furniture, natural objects, human-made objects, landscapes, etc. *Caltech-256* (Griffin et al, 2007) is an extended version of Caltech-101 that contains 30607 images grouped into 257 object classes (256 object classes and one additional background class).

Sample images from the above-mentioned object image datasets are illustrated in Figure 2.12.

(a)

(b)

Figure 2. 12: Sample images from object image datasets: (a) COIL-100, (b) Caltech-256

3) Medical image datasets

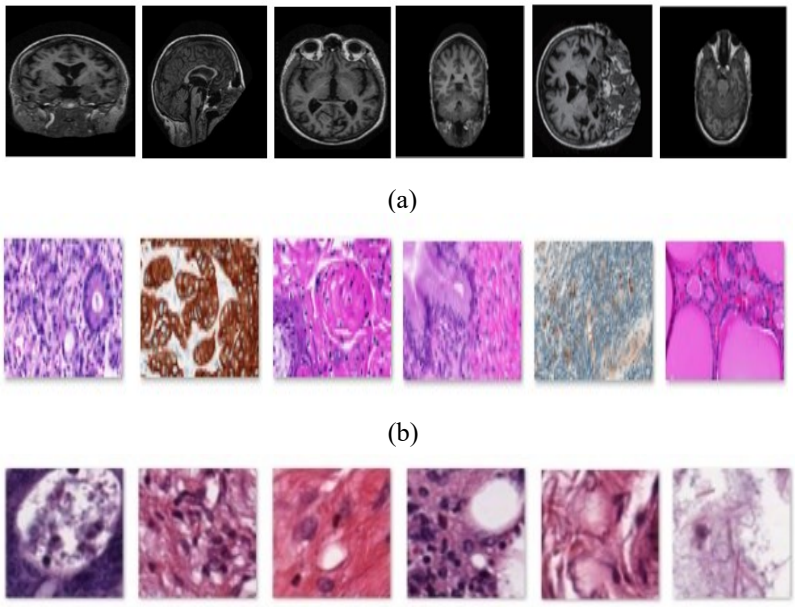
- **OASIS-MRI:** OASIS-MRI or OASIS-1 (Marcus et al, 2007) is a publicly available bio-medical image dataset developed by Image Studies Open Access Series (OASIS). It is a collection of MRI images that consists of 416 brain scans with a resolution of 208×208 pixels. These images, which are obtained from subjects between 18 and 96 years, are grouped into four classes based on clinical dementia rating (CDR) scale.

The latter is a clinical measure commonly used in Alzheimer's research and diagnosis to assess the level of cognitive impairment of individuals.

- **KIMIA Path960:** This dataset is primarily designed for research in digital and computational pathology (Kumar et al, 2017). In particular, it is widely used in tasks such as image classification and CBIR (Kumar et al, 2017; Lan & Zhou, 2016; Sukhia et al, 2019). KIMIA Path960 is a collection of histopathology images from 20 diverse tissue types such as muscle, epithelial and connective tissue. Each type of tissue or class contains 48 images of a resolution of 308×168 pixel, forming a total of 960 images in the database.

- **Kather-5K:** It is a colorectal cancer histology image dataset designed for histopathological tissue analysis. It contains 5000 histological images from eight different classes. Each class of Kather-5K represents a specific type of tissue textures of human colorectal cancer and includes 625 images of 150 × 150 pixels. TUMOR (tumor epithelium), STROMA (homogeneous tissue), MUCOSA (normal mucosal gland), COMPLEX (single tumor cells or immune cells) and EMPTY (histological image background) are examples of these classes.

Sample images from the above-mentioned datasets are shown in Figure 2.13.



(a)

(b)

(c)

Figure 2. 13: Samples from medical image datasets: (a) OASIS-MRI, (b) KIMIA Path960, (c) Kather-5K

2.8 Practical applications of CBIR

As outlined above, CBIR allows to index and search images from massive databases with reduced time and less manual effort. That is why it has managed to make inroads into numerous domains, including medicine, agriculture, crime investigation (i.e., Person Re-IDentification), e-commerce product retrieval, Internet of Things (IoT), Geographical information and remote sensing, fashion and interior design, etc. In this section, we discuss some common practical applications of CBIR, with a particular attention to the medical field where the use of CBIR techniques is continuously increasing.

- **Medicine:** The significant developments in medical imaging techniques (such as X-ray, computer tomography (CT) and magnetic resonance imaging (MRI)) along with the notable advance in the communication and data transfer protocols used in hospitals and clinics (such as digital imaging and communications in medicine (DICOM) and picture archiving and communication system (PACS)) have led to the generation of enormous amounts of digital images, and therefore the formation of massive medical image databases. PACS (Huang, 2011) is a medical imaging technology used for storing, retrieving, distributing and presenting medical images. Mainly, it allows the creation of large digital repositories where medical images are stored, and facilitates the access to these repositories by healthcare professionals over a network. PACS is a text-based search system that has been considered as a valuable tool for researchers and clinicians by offering more opportunities for research and disease diagnosis based on the comparison between different patient images (Kumar et al, 2013). Image comparisons can serve as support for a diagnosis or a treatment of a given patient by analyzing images from different patients and searching those that have similar characteristics to the patient image. However, like all text-based image retrieval systems, the search mechanism of PACS relies on textual keywords such as patient names, identifiers, image device, etc.; resulting in the frequent problems posed by this type of search (Kumar et al, 2013). One of these problems is the manual annotation which is impractical for the massive volume of medical images stored in the digital repositories. Another problem of PACS is that its search capabilities are limited by the text descriptions (Kumar et al, 2013). In fact, users often need to refer to the clinical reports or must have knowledge of specific keywords to find the desired images. Which can be inefficient and time-consuming especially if the clinical staff is not already acquainted with the identifiers and characteristics of those images. To enhance image retrieval efficiency and effectiveness of medical images, the

comparison between images of different patients need to rely on the visual characteristics of images rather than the associated text descriptions. In this context, Content-Based Medical Image Retrieval (CBMIR) is currently considered as a prominent tool for Computer-Aided Diagnostic (CAD), which can support clinicians in the diagnosis and treatment of many diseases (Das & Neelima, 2017). CBMIR systems allow indexing and retrieving similar medical images based on their visual representation.

In the literature, many CBMIR approaches have been proposed for different image types and clinical applications. Indeed, these approaches have dealt with different types of medical images, including radiological images such as X-rays, MRIs and CT scans, biomedical images such as histology and histopathology images, etc. The first CBMIR systems (Kelly & Cannon, 1994), have been developed in the early 1990s. Actually, in these systems, the retrieval of medical images was incorporated as a subdomain of the traditional CBIR system to trial and test the effectiveness of the latter when dealing with medical images. Trials and experiments performed on these systems have shown that the efficiency of these early systems was not as expected (Das & Neelima, 2017). Afterward, in order to ensure more effective image retrieval for healthcare applications, CBMIR systems like ASSERT (Shyu et al, 1999), IRMA (Keysers et al, 2003), BRISC (Lam et al, 2007), FIRE (Deselaers et al, 2005) were introduced as specialized medical image retrieval systems to handle the specific requirements of the medical domain and the sophistication of medical images. In recent years, CBMIR has undergone substantial progress, harnessing advancements in computer vision techniques, especially deep learning. In (Aher & Lilhore, 2016), an improved CBMIR architecture based on fuzzy c-means clustering and modified SVM classifier has been proposed to efficiently and effectively retrieve brain tumor MRI images. In (Kapadia & Paunwala, 2021), a Multi-Channel CNN (MC-CNN) model has been proposed to extract multiple high-level features from medical images. Specifically, the multiple feature vectors for the input images have been generated through the three channels of the MC-CNN model. Then, a concatenation layer has been used to combine these multiple features. In (Souid et al, 2023), a CBMIR system based on deep learning techniques has been proposed to efficiently retrieve chest radiograph images and assist radiologists in diagnosing complex medical cases. To extract relevant features, YOLOv5 and EfficientNet models have been used. In fact, YOLOv5 model has been used due to its effectiveness and efficiency in object detection; and EfficientNet-B0 and -B3 have been utilized for the image retrieval task and the prevention of noisy predictions, respectively.

- **Agriculture:** Content-Based Image Retrieval (CBIR) has been notably applied in the agriculture sector in order to support plant biologists and agricultural farmers in detecting and diagnosing plant diseases (Hanif et al, 2022). Generally, plant diseases can be perceived from the physical and color changes of the affected plant components such as roots, leaves and stems. However, due to the large variety of plant diseases and crops that can affect the different parts of plants; the detection of plant diseases based on the physical analysis of the signs and symptoms occurred on the plant components may be difficult even by qualified agricultural experts and plant pathologists (Singh & Yogi, 2023). In this context, CBIR can assist agricultural professionals and plant biologists by providing automated and efficient tools for an accurate and quick diagnosis of plant diseases, and consequently minimizing crop losses (Hanif et al, 2022). Many CBIR approaches have been proposed in the literature to deal with plant images such as (Gurubelli et al, 2023; Mujawar et al, 2014; Raja & Karthikeyan, 2022).

- **Fashion image retrieval:** Due to the increasing demands of the fashion industry, fashion-related research has attracted considerable attention from the computer vision researchers in recent years. Indeed, the expansion of online shopping with the extensive use of social media platforms like Instagram and Pinterest has led to the accumulation of a huge number of fashion images including clothes, footwear and accessories. In order to manage these huge number of images and search for specific items from the large collections of fashion images, content-based fashion image retrieval (CBFIR) has been usually employed in many e-commerce systems and some search areas such as Google, Taobao, Baidu, etc. (Shoib et al, 2023). In particular, the task of retrieving fashion items in CBFIR must be carried out across two domains of images, user-generated images and seller-generated images (Li et al, 2021b). Thus, the main challenge in CBFIR is to develop advanced methods and techniques that can deal with the two types of images generated from two different sources (Li et al, 2021b). In fact, images generated by users (i.e., customers) might be taken by themselves in uncontrolled environments; in contrast, images generated by professionals (i.e., sellers) are usually taken with controlled lighting and background. In addition, clothing items are usually deformable (i.e., might look different from one image to another depending on how they are worn or how they are designed), and their viewpoints are extensively changeful (i.e., the perspective of the clothing item might differ from the user image to the seller image). To overcome these challenges, many computer vision techniques have been leveraged to achieve efficient and effective image retrieval across the two domains of fashion images (Gupta et al,

2018; Li et al, 2019; Tariq et al, 2019; Varshney et al, 2024).

- **Person re-identification:** Also known as person retrieval, person re-identification is a specific application of CBIR. It consists in matching images of the same person that are taken by cameras with different views (Li et al, 2021b). Accordingly, several challenges face person re-identification; among them, the significant differences between the images captured on distinct camera views (Li et al, 2021b). That means that images taken by different cameras for the same individual may vary significantly from one image to another depending on different factors such as the background and the appearance. Another challenge is the interference of similar images with distinct identities (Li et al, 2021b). In other words, sometimes images of different persons may appear quite similar, causing difficulty in distinguishing between them. In addition, variations in the individual pose and occlusion (i.e., body covering) can make the retrieval task more difficult (Li et al, 2021b). To overcome these challenges, many CBIR approaches have been proposed to enhance the person retrieval performance. These approaches can be categorized into two groups, handcrafted and deep learning methods (Dou et al, 2022). Among the handcrafted methods, we cite the works of Liao, S. et al. (Liao et al, 2015) , Li et al. (Li et al, 2021a) and Liang et al. (Liang et al, 2024).

- **Internet of Things:** IoT has gained widespread use over the last years due to its capability to integrate smart systems, smart objects, frameworks and detectors into our daily lives (Hanif et al, 2022). It enables connecting electrical devices and sensors through internet. Usually, these smart devices generate a vast amount of sensitive data including images. In order to manage this sensitive data and offer an expanded storage capacity, many companies such as Amazon, Google, and Microsoft have leveraged Cloud computing technology to provide its customers with more economical services (Abduljabbar et al, 2019). However, the storage of the sensitive data, especially private or confidential images, on untrusted Cloud servers may lead to the violation of the images privacy (Abduljabbar et al, 2019). In order to secure sensitive images from unauthorized access, images are usually encrypted before being uploaded to the Cloud servers. Despite its ability to protect sensitive images, encryption process makes the task of searching specific images from the encrypted images more difficult (Abduljabbar et al, 2019). To overcome this challenge, CBIR methods have been used to efficiently and effectively manage and search the encrypted images based on their visual content without exposing this content to unauthorized access (Abduljabbar et al, 2019; Hsu et al, 2012; Lu et al, 2009). These methods have allowed to retrieve images from encrypted databases while preserving their confidentiality using various searchable encryption schemes

such as Min-Hash and order preserving encryption (OPE) in (Lu et al, 2009), homomorphic encryption in (Hsu et al, 2012) and matrix multiplications based encryption method in (Abduljabbar et al, 2019).

2.9 Conclusion

This chapter has provided an extensive review of CBIR. It has traced the evolution of image retrieval systems while discussing the prominence of CBIR systems over TBIR and SBIR. It has also explored the principal components of CBIR, with a particular emphasis on feature extraction. The chapter has also presented various matching measures, common performance evaluation metrics and benchmark datasets used in CBIR, and some of its applications. Essentially, it has been highlighted in this chapter that the choice of the appropriate feature extraction method and matching measure metric directly impacts the CBIR performance. In particular, local binary pattern method and its variants have been a preferred choice among many researchers for feature extraction and are still extensively used in current studies. In the next chapter, we will delve deeper into these methods.

Chapter 3

Local Patterns-Based Methods for Image Characterization

3.1 Introduction

As mentioned in the previous chapter, texture is a powerful visual feature for image retrieval. Among the effective and discriminant methods used for extracting texture features are local pattern methods. In this chapter, we provide a profound exploration of local pattern methods, with a special emphasis on LBP method and its color variants. We start by introducing the principal concepts of local pattern methods, highlighting their effectiveness in texture analysis and image retrieval. Next, we examine the basic LBP method and its variants, outlining their strengths and limitations. Following this, we discuss existing techniques for optimizing local pattern features, including pattern selection and pattern weighting.

3.2 Overview

Texture analysis holds a key position in computer vision and image processing applications (Zhou et al, 2008). Research on texture analysis has been ongoing since the 1960s (Mäenpää, 2003). Indeed, the importance of texture analysis stems from the importance of the texture itself. The latter is a vital element to the human vision as it provides useful information that helps humans understand the depth of scene and the orientation of surfaces (van den Broek et al, 2008). Naturally, the human eyes can easily distinguish between different textures. For instance, it can perceive the smoothness and roughness of surfaces as well as the regularity of patterns on objects or surfaces. Thus, the aim of research on texture in computer vision is essentially to simulate and mimic the human visual capability of learning and interpreting textures, while relying on the computer technologies (Dixit & Hegde, 2013). For that, research focus is to understand, process and represent texture using effective models (Dixit & Hegde, 2013). In CBIR, texture is a fundamental element which is used as a distinguishing

feature for describing images and facilitate their matching. Typically, textures within an image are constituted of repeating patterns that cover most or the entire image area (Muslihah et al, 2020). In this context, local pattern methods such as LBP can efficiently detect the different micro patterns of an image, such as edges, constant areas and points, what makes local pattern methods among the best and most popular methods used today for describing texture content within images (Ahmed et al, 2014; Takala et al, 2005). These methods capture texture information by identifying local patterns within images and their spatial relationships. In the following, we outline the advantages and limitations of these features.

3.2.1 Advantages of local pattern methods

Local pattern methods are a worthy choice for CBIR systems, especially in environments with limited resources (Arora & Sharma, 2024). In general, the advantages of local patterns can be summarized in the following points (Arora & Sharma, 2024):

- They require low computational and memory resources, making them an ideal solution for situations where the computational resources of the system (like CPU/GPU) are limited or its memory is constrained (Arora & Sharma, 2024).
- They are highly effective for systems with limited data or specialized applications. For instance, local pattern methods are very suitable for histopathology images because these latter are particularly comprised of cellular and tissue textures of repetitive patterns (Cai et al, 2022; Khadilkar, 2022). In addition, histopathology image datasets are generally of small size and limited training set data, making their analysis more challenging (Ayyad et al, 2021).
- They require reduced necessity to high-volume training data. In fact, models based on local pattern methods can perform well without needing a vast amount of training data (Arora & Sharma, 2024).
- They provide quick decisions and fast system response, which makes them appropriate for systems that require low-latency response (minimal delay) or real-time applications (Arora & Sharma, 2024).

3.2.2 Limitations of local pattern methods

Despite their advantages, local pattern methods have some limitations which can be addressed by different ways, ranging from a simple pre-processing step to more advanced

techniques (such as feature reduction) (Nava et al, 2012; Yazdi & Erfankhah, 2020). These limitations can be summarized in the following points:

- They are sensitive to noise (Nava et al, 2012): Local pattern methods are highly sensitive to noise, particularly if the descriptor is of a small neighborhood and hence a small amount of information is associated to a pixel. This, in turn, can reduce the texture representation accuracy.
- They show limited illumination invariance (Nava et al, 2012): Local pattern methods, and in particular LBP, can only handle monotonic illumination changes (simple changes in brightness across the entire image that do not affect the relative differences between the intensities of the image pixels). Therefore, they are generally less effective under extreme lighting conditions (such as strong shadows, highlights, reflections, etc.) which can affect the appearance of textures in images.
- They are characterized by high dimensionality (Ramola et al, 2020) : Some local pattern methods can result in high-dimensional features. This can arise from the size of neighborhood. In fact, a big number of pixel neighbors can increase exponentially the feature size, increasing the computational complexity in terms of both space and time**.**

3.3 Local Binary Pattern

Local Binary Pattern (LBP) is a mainstay texture method in computer vision. Renowned for its simplicity, robustness and efficiency, LBP has remarkably conquered various fields with outstanding success. Included in these fields are domains such as medical image analysis (Camlica et al, 2015), face and motion recognition (Elias et al, 2019; Liang et al, 2020), image and video retrieval (Takala et al, 2005; Zhao et al, 2011), texture classification (Ahmed et al, 2014), remote sensing (Zhang et al, 2022), etc. LBP was initially mentioned in the work of Harwood et al. (Harwood et al, 1995); then, it was publicly introduced by Ojala et al. (Ojala et al, 1996) as a texture analysis method. Ever since, LBP has rapidly become widely recognized and extensively applied in many real-time applications and high-profile events. In the 2008 Beijing Olympic Games, a face recognition system based on LBP was employed to verify the identity of a large crowd of visitors from the entire world (Ramola et al, 2020). LBP is a statistical method that captures the local structure of an image by comparing its pixel

intensities (Ramola et al, 2020). It can also be considered as a unifying approach to the statistical and structural methods (Pietikäinen et al, 2011) since it allows deriving the global statistical summary of the image texture by capturing the local structural patterns (such as edges, corners, and uniform regions) and their spatial relationships. The original LBP operator, developed by Ojala et al. (Ojala et al, 1996), works in a 3×3 neighborhood. In this regard, a label is assigned for each pixel in the image using the center pixel value of the 3×3 block as a threshold. To obtain the LBP labels, the intensity value of each neighbor is compared to the center pixel value, in a way that a binary value is assigned for each neighbor (see equation (3.1)) surrounding the center pixel. Then, the binary values of the neighbors are converted into a decimal value by multiplying each neighbor value by its corresponding weight. These weights are usually powers of 2, depending on the position of the pixel in the neighborhood; and since the neighborhood in the original LBP consists of 8 pixels, 2^8 (= 256) different labels (patterns) can be obtained for the 8-pixel neighborhood. After the conversion of the binary values of the neighbors, the obtained decimal values are summed up to get the decimal value of the center pixel which is called the LBP code. Figure 3.1 illustrates an example of construction of the original LBP operator. The original LBP operator is denoted by $LBP_{8,1}$; where 8 is the number of the neighboring pixels, and 1 is the radius of the neighborhood which forms a circle around the center pixel.

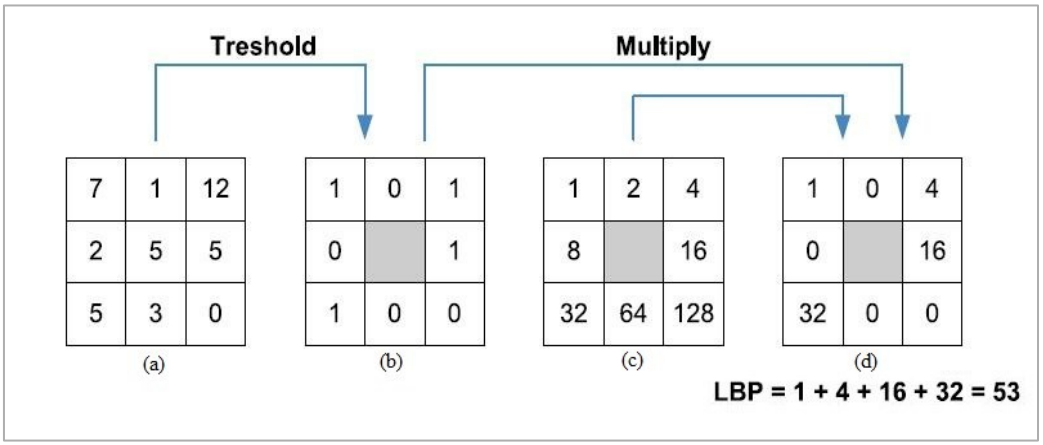


Figure 3. 1: An example of the original LBP operator construction: (a) 3 × 3 neighborhood, (b) binary pattern, (c) weights, (d) LBP code (Takala et al, 2005)

In general, the LBP operator is denoted by $LBP_{P,R}$, where P is the number of neighbor pixels and R is the neighborhood radius.

$$LBP_{P,R}(x_c, y_c) = \sum_{p=1}^{P} S(g_p - g_c)2^{p-1} \tag{3.1}$$

$$S(g_p - g_c) = \begin{cases} 1 & if \quad g_p \geq g_c \\ 0 & else \end{cases} \tag{3.2}$$

Specifically, the LBP operator has several key characteristics that made it a fundamental tool in computer vision and texture analysis. The invariance of LBP against monotonic gray level changes may be considered as the most important characteristic of LBP that make up its strength, especially in real-world applications where the illumination conditions can be unreliable (Pietikäinen et al, 2011). Another important characteristic of the LBP operator that has contributed to its widespread adoption is its computational simplicity, which allows an efficient image analysis even in challenging real-time settings where an immediate image processing and response are important (Pietikäinen et al, 2011). As a texture descriptor, LBP has an important characteristic over other descriptors which is its high discriminative power (Ramola et al, 2020). Indeed, LBP can effectively capture the texture information from images, distinguishing between various textures and patterns. Another key characteristic of the LBP operator is its flexibility or adaptability (Pietikäinen et al, 2011). In fact, LBP has been successfully employed in several applications and for several types of problems.

Despite all of those characteristics and advantages, the basic LBP, like any method, has a few limitations that have prompted researchers to develop other variants and extensions to overcome its limitations and improve its performance. Sensitivity to noise is one of the limitations of LBP that can affect its performance (Tyagi, 2017). Another limitation of the basic LBP is the rotation invariance issue (Ramola et al, 2020). In fact, the LBP operator is not invariant to rotations which makes it less robust in applications where rotational invariance is essential, such as face and motion recognition. Another issue can appear when the number of neighbors is increased. It is in fact the issue of high feature size which results from a high number of neighbors, thereby leading to a higher computational complexity whether in terms of time or memory space. In addition to the above, The LBP operator captures a limited structural information (Ramola et al, 2020). It can capture only information about the local structure of the texture in images and ignores the magnitude information (large-scale textural structures).

To address the above-mentioned limitations of the original LBP and enhance its robustness, several extensions and modifications of the latter have been introduced over the years.

3.4 LBP variants

In this section, we present some of the most prominent and widely used variants of LBP that were proposed to overcome certain limitations of the original descriptor or to improve the system performance in specific applications.

3.4.1 Grayscale-based LBP variants

Since its introduction, many variants of the original LBP have been proposed in the literature. Traditional LBP variants have been mainly designed to handle grayscale images, relying on intensity values (gray levels). Among these variants, we mention the following:

3.4.1.1 Dominant LBP

With the aim of reducing the LBP feature dimension while maintaining important textural information, Dominant LBP (DLBP) method (Liao et al, 2009) extends the original LBP by focusing on the dominant patterns (i.e., the most frequently occurred patterns) in an image. Although the uniform LBP method extracts fundamental texture information and reduces the LBP feature dimension by considering only the uniform patterns, these latter may not be the dominating patterns in certain textures that exhibit irregular edges and shapes (Liao et al, 2009). DLBP patterns, in contrast, are more informative and represent the most important texture information in the image.

The idea of DLBP is to select the patterns that contribute the most to the overall texture description. Liao et al. (Liao et al, 2009) have demonstrated that a minimum subset of patterns that constitutes about 80% of the total pattern occurrences in the image can capture the most important texture information. This subset of dominant patterns is therefore selected and the remaining less frequent patterns are ignored. The latter are generally the patterns that represent noise or less important information. Hence, the feature dimension is significantly reduced while capturing discriminative texture information.

Considering a neighborhood of m pixels at a radius R, the computation of the number of dominant patterns, covering about 80% of the total pattern occurrences in an image I, is as follows (Liao et al, 2009):

1. Initialize the number of dominant patterns (K_{temp}=0).
2. For each image I in a given training image set, initialize a pattern histogram H (H[0…(2^m-1)]=0) that count the occurrences of each pattern in the image.
3. For each center pixel in I, compute its pattern label and increase its corresponding bin in H by 1 (H[$LBP_{m,R}$]++).
4. After computing the pattern labels of all pixels in I, sort the histogram H in descending order of frequency.
5. Determine the number of patterns k required to reach 80% of the total pattern occurrences in I, as in the following equation:

$$k = argmin_k \left(\frac{\sum_{i=0}^{k-1} H[i]}{\sum_{i=0}^{(2m-1)} H[i]} \geq 80\% \right) \tag{3.3}$$

6. Accumulate the number of patterns required across all the training set images: K_{temp}= K_{temp}+k.
7. Finally, compute the required number of patterns occurrences for the training image set ($K_{80\%}$) as follows:

$$K_{80\%} = \left[\frac{K_{temp}}{Number\ of\ training\ images} \right] \tag{3.4}$$

3.4.1.2 Improved LBP

To capture more detailed local texture information and improve the discriminative capability of the original LBP, Jin et al. (Jin et al, 2004) introduced Improved ILBP (ILBP). The latter is constructed on the same fundamental idea of the original LBP. Furthermore, ILBP provides a complete representation of the local texture by considering the effect of the center pixel in the computation of the binary pattern, contrary to the original LBP in which the center pixel is just used as a threshold for comparing its neighboring pixels to its intensity. The intensity of the center pixel is then specifically included in the computation of the ILBP code and the largest weight is assigned to it. In fact, the threshold used ILBP is the mean intensity value

of all the neighborhood including the center pixel. The ILBP code of a given pixel (x_c,y_c) can be defined as follows:

$$ILBP_{P,R}(x_{c,}, y_c) = \sum_{p=0}^{P-1} S(g_p - m)2^p + S(g_c - m)2^P \tag{3.5}$$

$$m = \frac{1}{P+1}\left(g_c + \sum_{p=0}^{P-1} g_p\right) \tag{3.6}$$

where m is the mean intensity value of all the pixels in the P-pixel neighborhood, and the function S is defined as in equation (3.2).

By including all the pixels in a 3x3 kernel, the ILBP histogram computes 2^9-1 = 511 bins (by definition, all zeros and all ones are the same, that is why 1 is subtracted from 2^9). The dimension of the ILBP feature vector thereby increase significantly compared to LBP vector which computes only 2^8 = 256 bins, leading to a higher computational complexity (higher memory storage and processing time).

3.4.1.3 Local Ternary Patterns (LTP)

In order to address the limitation of noise sensitivity that confronts the original LBP, Tan and Triggs (Tan & Triggs, 2007) proposed a three-valued operator named Local Ternary Patterns (LTP). Unlike LBP which encodes the difference between the center pixel and its neighbors into two values (0 and 1), LTP encodes pixel differences into three values (0, 1, -1). This ternary encoding makes the operator more adapted for noisy environments and more consistent in either smooth or high-textured image regions (Ahmed et al, 2014). Thus, a threshold t was introduced to reduce the noise sensitivity. Therefore, a zone around the center pixel is defined in a way that pixel intensities in a zone of width ± t are assigned a value 0, while those above +t and below – t are assigned the values +1 and −1, respectively. Knowing that the threshold t is a constant value defined by the user, the binary threshold function S used in LBP (see equation (3.2)) is therefore replaced by the function S', which is defined as follows:

$$s'(g_p - g_c) = \begin{cases} 1 & if \quad (g_p - g_c) \geq t \\ 0 & if \quad |(g_p - g_c)| < t \\ -1 & if \quad (g_p - g_c) \leq -t \end{cases} \tag{3.7}$$

Each ternary pattern is then divided into two parts, upper pattern (the positive part) and lower pattern (the negative part), as depicted in Figure 3.2. The binary threshold functions for the upper and the lower pattern are then calculated as follows:

$$S^h(g_p - g_c) = \begin{cases} 1 & if \quad s' = 1 \\ 0 & otherwise \end{cases} \tag{3.8}$$

$$S^l(g_p - g_c) = \begin{cases} 1 & if \quad s' = -1 \\ 0 & otherwise \end{cases} \tag{3.9}$$

The ternary upper pattern ($LTP^h_{P,R}$) and lower pattern ($LTP^l_{P,R}$)are computed as follows:

$$LTP^h_{P,R}(x_c, y_c) = \sum_{p=1}^{P} S^h(g_p - g_c)2^{p-1} \tag{3.10}$$

$$LTP^l_{P,R}(x_c, y_c) = \sum_{p=1}^{P} S^l(g_p - g_c)2^{p-1} \tag{3.11}$$

The two parts are finally concatenated to form the LTP feature vector, which has a dimension of 512 bins (2×2^8).

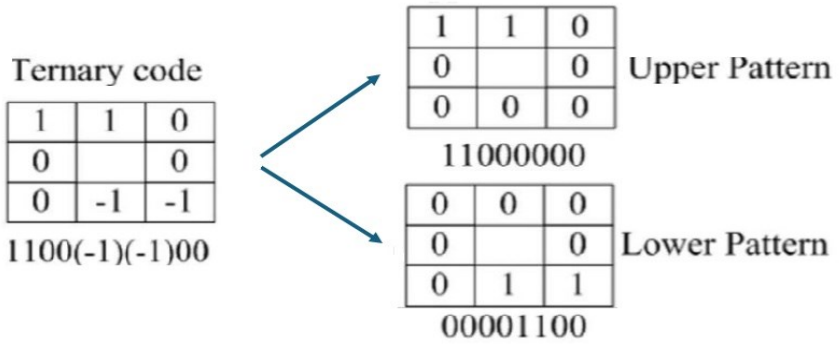


Figure 3. 2: An example of local ternary pattern operator (adapted from (Pietikäinen & Zhao, 2015))

3.4.1.4 Modified LBP

Modified Local Binary Pattern (MLBP) is an enhancement of the original LBP proposed by Naresh and Nagendraswamy (Naresh & Nagendraswamy, 2016) to improve texture classification of plant leaves. According to Naresh and Nagendraswamy, the original LBP may be unable to differentiate certain spatial structures such that it may generate the same LBP code for two different patterns having different spatial pixel arrangements. The reason of this limitation is the hard threshold of the original LBP, which is highly related to the difference between the center pixel and the neighboring pixels intensities (Naresh & Nagendraswamy, 2016). To improve the discrimination capability of LBP, an adaptive threshold is used in MLBP instead of the hard threshold of LBP. Specifically, the MLBP threshold is based on the mean (μ) and standard deviation (σ) of all the neighborhood. The mean and Standard deviation are local statistical measures calculated as follows:

$$\mu = \frac{\sum_{p=1}^{P} g_p + g_c}{(P+1)} \tag{3.12}$$

$$\sigma = \sqrt{\frac{\sum_{p=1}^{P} (g_p - \mu)^2 + (g_c - \mu)^2}{(P+1)}} \tag{3.13}$$

The MLBP code is defined as follows:

$$MLBP_{P,R} = \sum_{p=1}^{P} S * 2^{p-1} \tag{3.14}$$

where the function S takes the value 1 if the neighbor intensity is in the range [(μ-σ), (μ+σ)], and 0 if otherwise. The binary threshold S is defined as follows:

$$S = \begin{cases} 1 & if \ \ (\mu + \sigma) > g_p > (\mu - \sigma) \\ 0 & else \end{cases} \tag{3.15}$$

3.4.2 Color-based LBP variants

Due to their discrimination power, grayscale-based LBP descriptors have achieved great prominence in various texture analysis applications. However, as much as the texture information is important in describing images, the color information contained in images is also important. Furthermore, capturing texture or color information independently may not

discriminate images effectively, due to the fact that the human eye does not perceive texture and color separately (Agarwal et al, 2019). Thus, combining color and texture information in a unified descriptor can highly improve its discrimination capability (Agarwal et al, 2019). A simple way to construct a color LBP descriptor is to apply the basic LBP on the three-color channels of a given color space separately. Then concatenating the resulting three histograms into a single feature vector (such as LBP-RGB, LBP-HSV, LBP- LBP-Lab, etc). Besides, several extensions of the original LBP that incorporate the color information in different ways have been proposed in the literature, among which:

3.4.2.1 Opponent Color LBP

To capture both color and texture information, Opponent Color LBP (OCLBP) was proposed by Mäenpää & Pietikäinen in 2004 (Mäenpää & Pietikäinen, 2004). OCLBP is one of the earliest extensions of LBP that handles color images. The term "opponent color" used in OCLBP was defined according to the convention introduced by Jain and G. Healey (Jain & Healey, 1998) stating that all pairs of color channels are referred to as “opponent color” . In human vision, particularly, the opponent colors are those perceived as pairs of opposites (red-green and yellow-blue) (Pietikäinen et al, 2011). In computer vision, the concept “opponent color” is generalized to include any pair of color channels that are processed as opposing pairs (Pietikäinen et al, 2011). In this context, the OCLBP operator is constructed by applying LBP on each color channel independently and on pairs of color channels jointly. In the former, three color LBP histograms are computed for each color channel of the color space (the intra-channel features). In the latter, six color LBP histograms are computed for the three pairs of color channels (inter-channel features). In the RGB color space, the opponent-colors (R-G, R-B, G-R, G-B, B-R and B-G) of the three pairs of color channels are used to derive the six LBP histograms. However, as highlighted in (Mäenpää & Pietikäinen, 2004), opposing pairs, such as R-B and B-R, are highly redundant and consequently only one of them can be used while deriving the inter-channel features. As a result, the final OCLBP feature is constructed by the concatenation of the three intra-channel features (R, G, B) and the three maintained inter-channel features (R-G, R-B, G-B). Figure 3.3 illustrates the OCLBP computation scheme on the RGB color space.

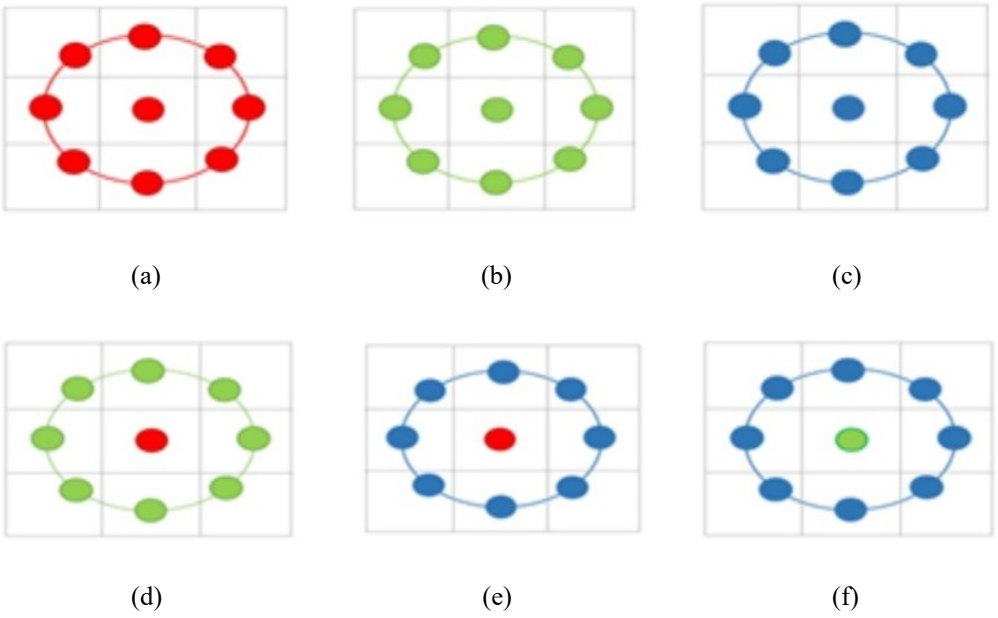


Figure 3. 3: OCLBP code computation for: (a) RR color channel pair, (b) GG color channel pair, (c) BB color channel pair, (d) RG color channel pair, (e) RB color channel pair, (f) GB color channel pair (Boukerma et al, 2025a)

The OCLBP code for a pair of channels M and N is defined as follows:

$$OCLBP(x_{c,M}, y_{c,M}) = \sum_{p=0}^{P-1} S(g_{p,N} - g_{c,M})2^p \quad (3.16)$$

where c is the center pixel, P is the number of the neighbor pixels surrounding it and R is the neighborhood radius. Knowing that the central pixel c is picked from the color channel M and its neighboring pixels are picked from the color channel N.

3.4.2.2 Improved Opponent Color LBP

Improved Opponent Color LBP (IOCLBP) (Bianconi et al, 2017) is a simple, robust and highly discriminative color-texture descriptor. In particular, IOCLBP was built upon OCLBP while considering the thresholding scheme of ILBP to enhance its invariance to illumination change. Thus, as in OCLBP, the IOCLBP feature is constructed by the concatenation of the intra- and inter-channel features; however, these latter are computed using the point-to-average thresholding scheme of ILBP instead of the point-to-point thresholding of LBP.

For a given color space of C channels and a neighborhood of n pixels (including the center pixel), the final IOCLBP involves C× (2^n-1) bins representing the intra-channel

features, and $K \times 2^n$ bins representing the inter-channel features (Bianconi et al, 2017). Given that K is defined as follows (Bianconi et al, 2017):

$$K = \frac{C!}{[2!\,(C-2)!]} \tag{3.17}$$

In contrast, the size of LBP features is only 2^{n-1} bins, and that of OCLBP consists of $(C+K) \times 2^{n-1}$ bins (Bianconi et al, 2017). IOCLBP produces then a high-dimensional feature compared to OCLBP, leading to higher space and time complexity. Nevertheless, when compared with other grayscale and color LBP variants on different types of image datasets, IOCLBP demonstrated great effectiveness and superiority over all compared variants across almost all datasets. In addition, it showed strong competitiveness compared to many pretrained CNNs, and great suitability for the classification of histological images.

3.4.2.3 Multichannel adder-based and decoder-based LBPs

To enhance the discriminative power of the original LBP and provide a robust texture representation for multichannel images, Dubey et al. (Dubey et al, 2016) proposed multichannel decoded local binary patterns based on adder and decoder mechanisms, which encode and combine LBPs from multiple color channels. Based on these two mechanisms, two color LBP descriptors were derived, namely multichannel adder-based LBP (maLBP) and multichannel decoder-based LBP (mdLBP). The two descriptors extend the original LBP to color images by using an encoding scheme incorporating spectral information across multiple channels (Mihoubi et al, 2018). In fact, to encode the relationships between the binary patterns of a multi-channel image, each pixel in the image is compared to its neighboring pixels over all the channels. Then, the sum of these binary comparisons across all channels is tested. After computing the LBP histograms in each channel for both operators, the histograms are then concatenated to form the final maLBP and mdLBP feature vectors. Figure 3.4 illustrates a flowchart representing the process of computation of maLBP and mdLBP from an image coded in the RGB color space (Dubey et al, 2016).

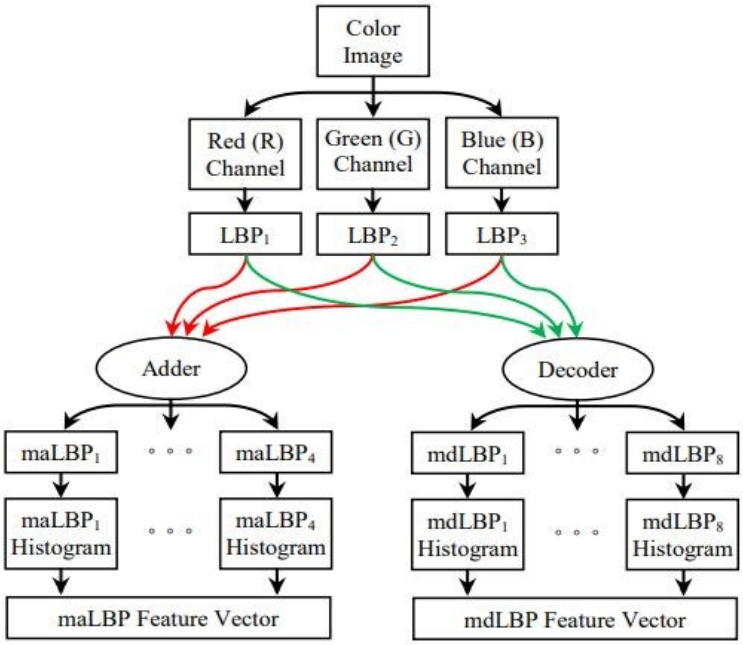


Figure 3. 4: Computation of maLBP and mdLBP of an RGB image (Dubey et al, 2016)

For a neighborhood of P pixels and a color space of C channels, the sizes of the final maLBP and mdLBP vectors are $(C + 1)\times 2^P$ and $2^C\times 2^P$, respectively (Mihoubi et al, 2018). Although both maLBP and mdLBP can capture effectively comprehensive texture and color information from color images, however mdLBP presents a major limitation which is the higher-dimensional feature space. Which in turn leads to a high computational complexity.

3.4.2.4 Multi-channel local ternary pattern

To effectively capture cross-channel color–texture information from images, Agarwal et al. (Agarwal et al, 2019) introduced Multi-Channel Local Ternary Pattern (MCLTP). The method leverages the advantages of LTP descriptor, which can effectively capture fine texture details, while capturing the color information from multiple color channels. More precisely, the LTP features are computed from each channel of the HSV color space; then concatenated into a single color-texture feature. In the computation of MCLTP, the V-channel is considered as a reference. So that, for each pixel in the image, the pixel from the V-channel that corresponds to it is considered as the center pixel. Then, a 5×5 neighborhood around the chosen center pixel is created from H, S, or V Channels, so that this neighborhood is selected from one of three channels alternatively. Subsequently, the LTP codes are computed for the 3 × 3 as well as 5 × 5 neighborhood, as in equations (3.18) and (3.19).

MCLTP patterns for 3 × 3 neighborhood and a radius R=1:

$$MCLTP_{p,R}^{C-V} = \begin{cases} 1 & g_{p,R}^{C} \geq g_{c}^{V} + t \\ 0 & \left| g_{p,R}^{C} - g_{c}^{V} \right| < t \\ -1 & g_{p,R}^{C} \leq g_{p}^{V} - t \end{cases} \quad (3.18)$$

MCLTP patterns for 5 × 5 neighborhood:

$$MCLTP_{p,R+1}^{C-V} = \begin{cases} 1 & g_{p,R+1}^{C} \geq g_{p,R}^{C} + t \\ 0 & \left| g_{p,R+1}^{C} - g_{p,R}^{C} \right| < t \\ -1 & g_{p,R+1}^{C} \leq g_{p,R}^{C} - t \end{cases} \quad (3.19)$$

Where C is a channel from the HSV color space; g_p^C is the neighbor pixel value for the channel C (C ∈ {H, S, V}), and g_c^V is the center pixel value for the channel V.

Finally, the final feature descriptor is constructed by combining the upper and lower pattern histograms obtained from the MCLTP values, across three combinations of channels, namely H–V, S–V and V–V. The MCLTP feature vector is of size 3072 bins. So, even if MCLTP is more robust against illumination changes and can capture fine texture variations while incorporating color information, it is computationally expensive and leads to a high-dimensional feature vector.

In addition to the LBP variants discussed in detail above, additional relevant variants are summarized in Table 3.1.

Table 3.1: Summary of some LBP variants

Ref.	*Name of the variant*	*Type*	*Fundamental principle*	*Motivation of extension*
(Ojala et al, 2002)	Uniform LBP (ULBP)	Grayscale	Consider only "uniform patterns" among all LBP patterns. The uniform patterns are those which have at most two bitwise transitions (from 0 to 1 or 1 to 0).	Reduce the feature size of the original LBP operator and improve the computational efficiency.
(Ojala et al, 2002)	Rotation Invariant LBP (RI-LBP)	Grayscale	Circularly rotating the LBP code in all possible directions until its minimum value is obtained. The latter is considered as the RI-LBP code.	Make the original LBP operator robust to rotation changes.

(Heikkilä et al, 2009)	Center-Symmetric LBP (CS-LBP)	Grayscale	Compare the intensities of center-symmetric pairs of pixels instead of comparing the pixels intensities to that of the center pixel.	Reduce the dimensionality of the original LBP and improve its robustness to illumination changes.
(Guo et al, 2010)	Completed LBP (CLBP)	Grayscale	Incorporate additional information such as sign (S), magnitude (M), and center pixel intensity information (C).	Enhance the robustness to illumination changes of the original LBP descriptor by exploiting all available information of the local structure of images.
(Vimina & Divya, 2020)	Maximal Multi-Channel LBP (MMLBP).	Color	Capture the color information from multiple channels by using an adder map to sum up the LBPs computed at each pixel location for all color channels. The MMLBP histogram is constructed by counting the frequency of occurrence of the different MMLBP values in the entire image.	Reduce the dimensionality of the maLBP and mdLBP descriptors.
(Babu et al, 2022)	Symmetric Inline Matrix-LBP (SIM-LBP)	Grayscale	Consider the intensity differences between the center pixel and its two closest left and right neighbors instead of considering the difference between the intensities of the center pixel and its neighbors.	Designed to be more robust to noise caused by low lighting conditions or blurry images.
(Turhal et al, 2024)	Multi-color multi-level LBP features (MCML_LBP)	Color	Extract LBP features at three levels of sub-blocks (1×1, 2×2, 4×4) from two color spaces (HSV and Lab), then, concatenate the obtained multi-level feature vectors for each color channel of the two-color spaces to form the MCML_LBP histogram.	Capture detailed texture information (by applying LBP at multiple levels) to enhance the discrimination and robustness of LBP.

3.4.3 LBP-based hybrid methods

A straightforward way to enhance the effectiveness of feature extraction and the performance of computer vision applications (such as image retrieval, classification, face recognition, etc.) is to combine LBP with other methods (Liu et al, 2017a). The LBP descriptor is thus commonly combined with other methods, whether they are other LBP variants, machine

learning or deep learning methods (Imdad et al, 2024; Karanwal, 2024; Saminathan et al, 2024).

3.4.3.1 Combining different LBP variants

In (Karanwal & Diwakar, 2023), Karanwal and Manoj proposed Triangle and Orthogonal LBP (TAO-LBP) descriptor which combines two LBP variants, namely Triangle LBP (TLBP) and Orthogonal LBP (OLBP). TLBP features were derived by applying the LBP operator in vertical and horizontal directions using triangular image patches of two sizes (5 × 3 and 3 × 5). As for OLBP, the features were extracted by focusing on the orthogonal positions of the image patches. Thereafter, the two descriptors were combined into a single robust descriptor (TAO-LBP). In (Karanwal, 2024), a hybrid framework by combining different LBP variants was proposed for feature extraction. In particular, MRELBP-NI (Liu et al, 2016), MB-LBP (Liao et al, 2007) and RD-LBP (Liu et al, 2012) LBP variants were combined in a hybrid color space to extract robust and discriminative features. More specifically, three color channels were taken from the RGB, YCbCr and YIQ color spaces so that MRELBP-NI was extracted from the R channel of RGB, MB-LBP from the Cr channel of YCbCr and RD-LBP from the Q channel of YIQ. These variants extracted from the hybrid color space are then combined to form the final descriptor called RADLCP (Karanwal, 2024), with a feature size of 768 bins.

3.4.3.2 Combing LBP with other handcrafted methods

In (Nguyen et al, 2009), Nguyen et al. proposed a hybrid method called Local Gabor Binary Pattern Whitened Principal Component Analysis (LGBPWP) to extract robust features. In a first step, the authors combined LBP with Gabor filters to capture the local texture and global structural information from facial images. Then, they used a Whitened Principal Component Analysis (WPCA) scheme to reduce the dimension of the combined features. The whitening process in WPCA was used to normalize the reduced features obtained after the application of PCA. In (Liu et al, 2017a), Liu et al. proposed an hybrid feature extraction technique based on Color Information Feature (CIF) and LBP descriptors. In fact, to describe the global color distribution of images, the CIF descriptor (which is in turn was derived from the color histogram) was used for extracting the color features. On the other hand, to capture the local texture information, the LBP descriptor was used for extracting the textural features from images. The color and texture features were then combined to achieve a good performance for color image retrieval and classification. In (Saminathan et al, 2024), a hybrid method

called Discrete Wavelet Coefficients-Bag of Visual Words-Contour-LBP (DWC-BoVW-Contour-LBP) was proposed to extract robust and discriminative features, enhancing the performance of medical image retrieval. In particular, the LBP descriptor was used to extract the textual features along with Discrete Wavelet Transform (DWT) which was used to extract the frequency features, contour analysis to extract the shape features and Bag of Visual Words (BoVW) to extract the visual features. All the extracted features were subsequently combined using pixel-level image fusion to provide a more comprehensive feature representation of images.

3.4.3.3 Combining LBP with deep learning methods

In recent years, the LBP method combined with deep learning models (such as CNNs) has been extensively used to improve the performance of various applications. In (He et al, 2018), the LBP features were combined with GLCM and deep features to effectively describe the histopathology images. In fact, both LBP and GLCM were used to extract discriminative texture features. Furthermore, transfer learning was used for AlexNet (Krizhevsky et al, 2012) and GoogLeNet networks to extract deep features. The latter were refined to obtain a relevant subset of them. To leverage the strengths of handcrafted and deep learning methods, texture and deep features were then combined into a single feature vector. In (Ke et al, 2018), LBP was combined with CNN in a different way. Indeed, the LBP descriptor was first used to obtain the LBP coded images. Then, the coded images were normalized and subsequently employed as an input of a proposed CNN. The latter which was composed of nine layers was trained using the LBP coded images as an input. In (Imdad et al, 2024), the LBP method was used alongside other handcrafted and deep learning methods to extract a richer feature representation that could improve the CBIR performance. In particular, the LBP features were combined with Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005), Bag of Features (BoF), and SURF handcrafted features, in addition to the deep features extracted using a modified AlexNet. The AlexNet model was indeed combined with a Spatial Pyramid Pooling (SPP) layer to allow pooling features across multiple spatial levels and therefore handling images of different dimensions (without requiring to resize them). In fact, the LBP features fused with the other handcrafted features contributed to enhancing the discriminative power of the feature representation. On the other hand, the deep features contributed to describing the semantics of images by capturing complex patterns. By combining the

handcrafted and deep features, the overall feature representation was enhanced, leading to an improved performance of the CBIR system.

Further LBP-based hybrid methods, proposed in the context of CBIR, are summarized in Table 3.1.

Table 3.2: Some LBP-based hybrid methods used in the context of CBIR

Ref.	*Combined features*	*Type of combination*	*Dataset used*	*Strengths*	*Limitations*
(Yuan et al, 2011)	LBP SIFT	LBP with other handcrafted methods	Corel-1k	Capture discriminative texture information and specific object structures even in noisy image backgrounds or ambiguous objects.	Lower performance and discrimination in patch-based scenario.
(Ghahremani et al, 2021)	HOG Color histogram LBP	LBP with other handcrafted methods	Liver CT scan images	Leverage the strengths of color, texture, and shape visual features to provide a comprehensive feature description.	High-dimensional feature vector.
(Jiang, 2021)	HSV histogram LBP DTCWT AlexNet features	LBP with deep learning methods	Corel-1k	Improve the overall CBIR performance	Relatively complex and time-consuming.
(Khan et al, 2023)	LBP DWT Tamura	LBP with other handcrafted methods	Brodatz and MIT-VisTex	Extract finest texture information.	Increased computational complexity.
(Akçiçek et al, 2025)	HOG LBP Darknet53 and Densenet201 features	LBP with deep learning methods	Shoulder MRI images	Leverage the strengths of deep features and traditional texture features to efficiently detect acromion types in shoulder MRI images.	Small MRI image dataset. Single-center study with limited generalizability.
Kumar & Murthy (2025)	Color histogram Color correlogram Color moments GLCM LBP Gabor Filters	LBP with other handcrafted methods	Collected dataset of diverse images from different domains.	Exploit the complementary advantages of color and texture information to provide a robust and richer image representation.	High-dimensional feature vector. Sensitivity to illumination variations.

3.5 LBP-based feature selection

It can be observed from the local pattern methods reviewed in the previous sections that a considerable proportion of them, especially color and combined LBP methods, struggle with the issue of high-dimensional feature space. To address this issue, many feature dimensionality reduction methods have been proposed to reduce the dimension of the features extracted using LBP-based methods. Many studies have revealed that the LBP dimension can be reduced in two ways, either by reducing the number of pixels in the neighborhood or by selecting a subset of the LBP patterns (Pietikäinen & Zhao, 2015). Moreover, it has been further revealed that in many applications, a carefully chosen subset of LBP patterns can significantly improve the performance than the use of the entire set of patterns (Pietikäinen & Zhao, 2015). Overall, the goal of reducing the LBP based-features is to provide a more compact and discriminative representation by eliminating the redundant information while keeping the most important features (Huang et al, 2011). In addition, when using LBP-based methods in real-time applications, it is recommended to use an LBP- based feature vector of a reduced size to speed up the image processing and reducing the response time (Huang et al, 2011). In this context, LBP feature selection (also called pattern selection (Mehta & Egiazarian, 2016), bin selection (Nanni et al, 2012), (Porebski et al, 2018) or attribute selection (Maturana et al, 2011)) has gained a significant attention in the literature, and many techniques have been proposed to overcome the high-dimensional LBP issue (Liao et al, 2009), (Shan, 2012), (Wadhera & Agarwal, 2022). These techniques were based either on some rules, such as uniform patterns, or on some machine learning techniques, such as boosting learning and subspace learning. (Huang et al, 2011). Rule-based methods are simple and computationally efficient as they use predefined rules to select the patterns; however, they exhibit insufficient feature selection capacity. In contrast, LBP-based feature selection methods which use machine learning techniques have superior feature selection ability; however, they are computationally expensive because they typically require an offline training (Huang et al, 2011).

3.5.1 Rule-based selection

Uniform patterns is one of the most used rules for selecting the LBP features (Huang et al, 2011). In (Mäenpää et al, 2000), it has been demonstrated that only a small carefully selected subset of patterns can significantly contribute to the overall discriminative power and

robustness of the extracted features, the other patterns are less informative or redundant. In (Lahdenoja et al, 2005), the authors applied a symmetry level framework to the uniform patterns for further optimizing the patterns and thereby reducing their length. In particular, the LBP patterns were selected based on a symmetry level rule which allows maintaining the patterns with a high symmetry level (more discriminative and informative patterns) and discard those with a low symmetry level (less informative patterns). In (Liao et al, 2009), a simple rule that consisted in selecting the most frequently occurred patterns were used to derive the DLBP features. In fact, Liao et al. have demonstrated that only a subset of about 80% of the total image patterns can capture the most important texture information. This subset of dominant and discriminative patterns was therefore selected and the remaining less informative patterns were excluded.

3.5.2 Boosting LBP features

Boosting techniques are widely used for feature selection (Huang et al, 2011; Pietikäinen & Zhao, 2015). Many LBP-based feature selection approaches have also exploit these techniques to select discriminative LBP patterns for various applications (Pietikäinen & Zhao, 2015). AdaBoost is one of these techniques that was widely used for learning the most relevant patterns and improving the application performance (Hasoon & Hassan, 2019; Zhang et al, 2004). Indeed, Adaboost is a machine learning algorithm which combine various weak classifiers into a stronger classifier (Chehrehgosha & Emadi, 2016) . Weak classifiers (or weak learners) are simple models that AdaBoost weight and combine to boost their performance (Chehrehgosha & Emadi, 2016). G. Zhang et al. (Zhang et al, 2004) used AdaBoost to improve the face recognition performance by selecting the most discriminative patterns and their weights across different local regions in the image. Chehrehgosha and Emadi (Chehrehgosha & Emadi, 2016) proposed a structured Adaboost algorithm to enhance the performance of a human face detection. In fact, they used LBP descriptor to extract robust features that can distinguish between facial and non-facial regions. Then, they used AdaBoost to combine the weak classifiers applied on different facial regions to form a stronger classifier that improved the overall face detection accuracy. Hasoon and Hassan (Hasoon & Hassan, 2019) used AdaBoost for face image retrieval. To achieve a robust and efficient CBIR system, they extracted discriminative LBP features from the different facial regions of the image, then a boosted process based on AdaBoost was employed to weight and combine the weak classifiers associated with the LBP features of the different facial regions. By combining best-

performing classifiers of the facial regions into a more accurate strong classier, the most discriminative LBP features were automatically prioritized and thus selected, making the feature size more reduced and more discriminative.

3.5.3 LBP subspace learning

The goal of LBP subspace learning methods is to reduce the LBP feature dimension by projecting the LBP features from a high-dimensional space to a lower-dimensional space while maintaining a compact and discriminative feature representation (Huang et al, 2011). Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are common subspace learning techniques that are widely used for reducing the high-dimensional LBP features. Chavda and Goyani (Chavda & Goyani, 2020) used PCA to reduce the LBP-based feature dimension, by removing redundant features, and the LDA to select the optimal features from the reduced feature set. To improve the CBIR performance, they proposed a modified version of the multi-scale local binary pattern (MS-LBP) descriptor (Xia et al, 2013) combined with color to extract color-texture features. In fact, to reduce the size of the extracted feature, they first applied PCA on each feature vector; then, they concatenated them into a single feature vector. Subsequently, they applied an optimal selection on the features of the combined vector using LDA to enhance its discriminative power. Wadhera and Agarwal (Wadhera & Agarwal, 2022) employed PCA and LDA to select more robust features from their proposed Multi-block Neighborhood Combination Pattern (MNCP) features. MNCP descriptor is an enhanced version of LBP that encodes three different combinations of intensity differences between the pixels of the local neighborhood to derive a local pattern descriptor more robust to noise. Although the MNCP features were more discriminative and more effective for improving the retrieval performance of biomedical images, they were high-dimensional. In order to reduce the MNCP feature dimension, the authors used PCA to transform the initial feature set into a new coordinate system and thus identifying the principal components maximizing the variance in the data. And hence, the most important features were maintained and the less informative or redundant ones were discarded. Thereafter, LDA was applied to further optimize and refine the feature vector by enhancing the inter class separability, leading to an accurate retrieval of biomedical images.

3.5.4 Other methods

In addition to the above-discussed pattern selection approaches, further methods based on other machine learning techniques have been proposed for different computer vision tasks. In (Shahamat & Pouyan, 2015), Shahamat and Pouyan used the Genetic Algorithm (GA) (Holland, 1975) to select the most discriminative bins among various LBP histogram bins extracted from different independent components of functional MRI images. The aim was indeed to improve the classification accuracy of schizophrenia. In (Sotoodeh et al, 2019), k-means algorithm was utilized to select discriminative LBP-based features that enhanced the effectiveness and efficiency of color image retrieval. In particular, k-means was used to select a subset of representative features among three concatenated feature vectors. The latter were extracted using three proposed color LBP variants, namely CRMCLBP_S, CRMCLBP_M and CRMCLBP_C. In (Vensila & Boyed Wesley, 2024), Vensila and Boyed Wesley proposed an optimization technique based on Adaptive Particle Swarm Optimization (APSO) algorithm (Zhan et al, 2009) to select optimal LBP features which enhanced their proposed authentication system. In fact, the LBP features were extracted from different regions in biometric images including face, fingerprint and finger vein. Then, the computed LBP histograms of each region were concatenated to form a single feature vector. To reduce the dimension of the feature space, APSO was applied on the concatenated LBP features to select the most relevant features and remove the redundant or less informative features.

3.6 LBP-based feature weighting

3.6.1 Feature weighting concept

In general, selecting representative features that effectively describe the output behavior of data is a crucial task in many computer vision applications. Nevertheless, in Feature Selection (FS), it is ordinarily assumed that all the selected features are of equal importance (Niño-Adan et al, 2021). But in fact, in real world issues, not all features contribute equally in representing the hidden patterns (Niño-Adan et al, 2021). In this context, Feature Weighting (FW) techniques are used to assign varying degrees of importance to the input features, prioritizing the features that contribute significantly in the overall representation of data. In fact, FW techniques assign appropriately different weights to the features that represent a given dataset. Assuming that X ($X \in \mathbb{R}^{n \times m}$) is a dataset comprising n samples, each represented by m features, FW is the process that allows obtaining a set of optimal weights

w_{ij} ($i \in \{1,...,n\}$, $j \in \{1,...,m\}$) that determine the relative relevance of the features (Niño-Adan et al, 2021). Usually, these weights are between 0 and 1 and their sum is equal to 1 ($\sum_{i,j} w_{i,j}=1$), and multiplied by each value $x_{i,j}$ of the dataset X (Niño-Adan et al, 2021). A high weight value indicates a high importance of the corresponding feature; a low weight value, in contrast, refers to a less relevant feature. In addition, if the weight value is equal to 0 ($w_{i,j}=0$), that means that the corresponding feature does not contribute in the representation of the data. This can be regarded as a feature selection factor such that the feature of weight 0 is irrelevant and can effectively be discarded (Niño-Adan et al, 2021).

3.6.2 Weighting techniques for local pattern methods

Feature weighting has been particularly applied to the LBP method and its variants for various tasks, such as image retrieval, classification and face recognition (Boukerma et al, 2019; Song et al, 2019; Sotoodeh et al, 2019; Truong et al, 2022). Feature weighting in the context of local pattern methods involves assigning varying weights to the features extracted using the LBP method and (/ or) its variants, enhancing their performance and discriminative power. Several LBP-based weighting approaches have been proposed for learning appropriate weights at different levels: features, blocks or patterns (Boukerma et al, 2019; Song et al, 2019; Wei et al, 2014). Feature weighting in the context of LBP may therefore be applied on (i) multiple LBP or LBP variants features extracted from the entire image, (ii) different block histograms extracted from several blocks or regions within the image, or (iii) individual local patterns encoded using the LBP descriptor or its variants.

In (Wei et al, 2014), multiple LBP feature vectors extracted from the same image at different scales were combined and weighted to increase the accuracy of face recognition. Specifically, the local features were derived using the LBP operator or any of its variants at various scales. The latter are natural numbers used to produce new radius values through the multiplication of the base radius value by the defined LBP operator. In this regard, features computed at small scales (representing fine textures) were assigned higher weights, and features computed at large scales (representing coarse textures) were assigned lower weights. In (Truong et al, 2022), a weighting statistical binary pattern approach was proposed to enhance the discriminative power of LBP and its robustness against illumination changes. In fact, different CLBP features were extracted for the sign and magnitude components along four different directions, then they were weighted based on a variance moment framework.

The weighted histograms were subsequently fused to form an efficient and robust feature representation.

In (Song et al, 2019), a block weighting scheme was introduced to improve the representation of local textures and, consequently, enhance the classification performance. By assigning varying weights to the image blocks, the method was able to emphasize the important regions of the image that had a stronger effect on the classification performance, while decreasing the impact of less informative regions that represent redundant or uncertain data. A set of LBP histograms were then generated for the weighted blocks, and then concatenated into a more discriminative feature vector. In (Gao et al, 2021), an adaptive weighting scheme was proposed to emphasize the most discriminative regions of face images. The images were first divided into equal blocks; then, various LBP histograms were computed for each sub-block. Thereafter, the LBP histograms of the sub-blocks were weighted and concatenated to form a robust feature vector.

In (Boukerma et al, 2019), a local pattern weighting mechanism was proposed to enhance the CBIR performance and thereby reduce the semantic gap. The weighting mechanism which was based on the differential evolution algorithm generated an optimal weight vector whose values were assigned to the local patterns extracted using the LBP method or one of its variants (ILBP, LTP and MLBP). By emphasizing the most important patterns, the weighted local pattern methods were able to outperform the non-weighted methods in terms of retrieval precision. In (Kumar et al, 2020), appropriate weights were assigned to the patterns extracted using the LBP method, based on the position of matching between the center pixel and it neighbors. Indeed, to take advantage of all hidden information present in the neighboring pixels, the latter were compared to the center pixel to determine a similarity path. In fact, if a neighboring pixel matched the center pixel, a similarity was mentioned and a weight was assigned to the pattern based on the position of the matching neighbor vis-à-vis the center pixel.

3.7 Conclusion

In this chapter, we have delved into the local pattern methods and their impact on texture analysis. We have provided an overview of the local pattern methods as well as their advantages and drawbacks. Furthermore, we have brought attention to the local binary pattern method as it is one of the fundamental methods in CBIR. Finally, we have shed light

on the selection and weighting techniques particularly exploited for optimizing the local pattern-based features. To further optimize the feature extraction process and other CBIR components, recent CBIR systems consistently rely on optimization and machine learning techniques. In the next chapter, we will turn our attention to these techniques, highlighting their role in CBIR.

Chapter 4

Optimization and Machine Learning Techniques for CBIR

4.1 Introduction

Optimization techniques hold a key position in many computer vision applications. In particular, they are commonly used in CBIR systems for improving their efficiency and effectiveness. On the other hand, the literature shows that modern CBIR systems have shifted towards adopting Machine Learning (ML) techniques to learn meaningful image representation and thereby improve the search accuracy (Arora et al, 2023; Gautam & Khanna, 2024; Kale & Mukhopadhyay, 2022; Shabir et al, 2025). In that context, this chapter investigates optimization and ML techniques that tackle crucial issues in CBIR such as the curse of dimensionality and feature representation. We start by examining the role of optimization and ML techniques in CBIR, highlighting how these techniques contribute to enhancing the different components of CBIR systems. Following that, we explore optimization algorithms and ML methods within the context of CBIR, analyzing their taxonomy and highlighting their impact on CBIR systems performance.

4.2 How do optimization and machine learning techniques contribute to CBIR systems enhancement?

Optimization and ML techniques can enhance the performance of CBIR in various ways. In fact, they play a crucial role in enhancing the key components of CBIR, such as feature extraction and representation, and matching measurement (Arora et al, 2023; Mathews et al, 2022; Mosbah & Boucheham, 2017). Various optimization and ML techniques have been thus employed to address the CBIR challenges through the enhancement of its main components.

A. Enhancing feature representation

Although handcrafted methods are successful in representing images, relying only on them cannot generally suffice to address the main challenges of CBIR, especially the semantic gap issue. Indeed, optimization and ML techniques demonstrated notable effectiveness in enhancing the representation of images by optimizing the image features through feature selection and dimensionality reduction approaches or by learning high-level representations (Adegbola et al, 2020; Gautam & Khanna, 2024; Subramanian et al, 2022).

— **Feature selection and dimensionality reduction:** Basically, feature selection methods are used for reducing the high-dimensional features resulting from the feature extraction phase. To achieve a lower-dimensional feature, a subset of the most relevant features is generally selected from the high-dimensional features. Indeed, an efficient CBIR system is highly related to the selection of the important features from the high-dimensional vectors (Shukla & Kanungo, 2020). Finding the optimal subset of relevant features that can effectively reduce the feature dimension is a challenging task. In fact, for a feature vector composed of N features, 2^N possible subsets of features are evaluated using exhaustive search methods to find the optimal subset, which is computationally expensive especially with a large number of features (Shukla & Kanungo, 2020). To find the optimal subset of features without examining all possible feature subsets, optimization techniques provide a practical solution to the exponential growth of feature selection complexity (Kushwaha & Welekar, 2016; Reddi & Enireddy, 2016; Subramanian et al, 2022). In this context, FS is considered as an optimization problem whose solution is to efficiently explore the search space aiming at finding the optimal subset of features (i.e., the optimal solution) with lower computational cost. Among the widely used optimization techniques for FS in CBIR are genetic algorithm (Kushwaha & Welekar, 2016; Zhao et al, 2008), PSO (Subramanian et al, 2022; Vamsidhar et al, 2017) and Cuckoo Search (Al-Abaji, 2019; Reddi & Enireddy, 2016). On the other hand, machine learning techniques are widely used for feature dimensionality reduction in CBIR. These techniques focus mainly on projecting the high-dimensional feature space into a lower-dimensional space, aiming at improving the retrieval performance and decreasing the computational costs. Among the frequently utilized ML methods for feature dimensionality reduction in the context of CBIR, we mention PCA (Adegbola et al, 2020; Belarbi et al, 2017), LDA (Janarthanam & Sukumaran, 2016; Wadhera & Agarwal, 2022) and deep learning methods (Kale & Mukhopadhyay, 2022; Petscharnig et al, 2017).

— **High-level feature representation:** To address the semantic gap issue arising from the use of low-level features for representing images, deep learning methods are particularly widely used in recent years to automatically learn high-level representations from images (Arora et al, 2023). Notably, CNNs are reported to be very efficient for deriving robust feature representations (Hameed et al, 2021; Mathews et al, 2022)**.** In fact, deep learning models can automatically learn complex feature representations from images (Mathews et al, 2022). These features (such as objects and scenes) are typically extracted from the deeper layers of CNN models (Mathews et al, 2022). High-level features can thus capture the semantic information that align better with the human perception of images. Which allows bridging the semantic gap and achieving high retrieval performance. Pre-trained CNNs such as ResNet, GoogLeNet, MobileNet, AlexNet and VGG are commonly used in CBIR (Ahmed et al, 2021; Arora et al, 2023; Gautam & Khanna, 2024; Shabir et al, 2025).

B. Refining and learning effective matching measures

The majority of retrieval systems, comprising CBIR systems, explicitly rely on distance or similarity functions intending to align the feature descriptors with the human perceptual resemblance, and reflect a subjective similarity rather than just a computational similarity (Arevalillo-Herráez et al, 2008). To achieve this objective, advanced matching measurement functions have been proposed by leveraging optimization and ML techniques. These techniques are generally used to learn weighted combinations of matching measures that enhance the image matching and thereby retrieval performance (Arevalillo-Herráez et al, 2008). In (Alsmadi, 2017), an efficient matching measure based on a metaheuristic algorithm called memetic algorithm (which is an enhanced version of genetic algorithm) was proposed for CBIR. In (Mosbah & Boucheham, 2017), a matching measure selection paradigm using Sequential Forward Selector (SFS) metaheuristic (Whitney, 1971) and based on relevance feedback was proposed to enhance CBIR performance. In (El-Naqa et al, 2004) two machine learning methods, namely neural networks and SVMs were used to learn an effective matching function which was capable of predicting the human perceptual matching and improving medical image retrieval. In (Ahmed & Ibraheem, 2024), a matching measure function based on the combination of three weighted matching measures was proposed for CBIR. The fused matching measure was refined using PSO and a neural network method.

C. Enhancing relevance feedback

Relevance Feedback (RF) is a commonly used technique that can improve the performance of CBIR systems by adapting the retrieval process according to the user preferences (Alrahhal & Supreethi, 2024a; Jafarinejad & Farzbood, 2021; Mosbah & Boucheham, 2014) . In particular, RF is a refinement stage that can be incorporated in CBIR systems to progressively enhance the retrieval results based on the user intervention (Hameed et al, 2021). The latter is in fact the user's assessment of the relevance or non-relevance of the returned images, which permits to adjust the retrieval strategy until meeting the user intent (Hameed et al, 2021). In this context, optimization and ML techniques are extensively used to adjust the retrieval strategy and refine the returned results based on the user feedback (Alrahhal & Supreethi, 2024a; Jafarinejad & Farzbood, 2021 ; Mahmood et al, 2022 ; Mosbah & Boucheham, 2014; Putzu et al, 2020). Optimization algorithms like PSO and GA are widely used in CBIR for enhancing RF. In (Jafarinejad & Farzbood, 2021), PSO algorithm was used to directly weight the extracted features based on the information derived from user feedbacks. In (Mahmood et al, 2022), PSO and GA algorithms were used to enhance RF through the augmentation of the positive sample set identified by the user. In (Raghuwanshi et al, 2012), an interactive GA-based RF method was proposed to find the combination of descriptors that describe more effectively the user perception of the resemblance between images. ML methods like SVM, KNN, decision trees and CNNs are also commonly used in CBIR to improve the RF capabilities. In (Alrahhal & Supreethi, 2024a), a modified KNN-based RF algorithm was proposed to improve the retrieval performance based on the user interactions. In (Oh et al, 2010), a RF strategy based on SVM was proposed for content -based medical image retrieval. In (Putzu et al, 2020), a CNN model was tuned and updated according to the user's feedback to enhance the retrieval performance. In (Bhattacharyya et al, 2015), a Decision Tree (DT) was incorporated in the RF process to iteratively refine the retrieved results based on the user feedback.

4.3 Optimization techniques

Nowadays, a large number of the optimization problems, stemmed from the modern applications, are NP-hard problems (Nondeterministic Polynomial time problems) that no algorithm in a polynomial time is recognized for solving them (Jourdan et al, 2009;

Nesmachnow, 2014). Solving this type of problems is challenging. This is mainly due to numerous factors, including (Nesmachnow, 2014):

- The high-dimensional search space that they involve.
- The hard constraints that they follow and which make the feasible solutions rare.
- They are multimodal or multi-objective problems with heavy computation functions.
- They are time-varying problems involving sophisticated mathematical functions.
- They deal with large amount of data.

The literature shows that, generally, two approaches have been considered to solve NP-hard problems: exact methods and approximate methods (Jourdan et al, 2009). Exact methods such as branch and bound methods, linear programming and dynamic programming are generally adopted for small problems (Jourdan et al, 2009) . Although these methods can lead to an optimal solution and prove its optimality, they are extremely time-consuming and not practical for large NP-hard problems. On the other hand, approximate methods are most frequently used for large problems, when exact methods fail in solving them (solution too slow or impossible) (Jourdan et al, 2009). Approximate methods, which involve heuristics and metaheuristics, can therefore solve large optimization problems efficiently but do not ensure obtaining the optimal solution (Nesmachnow, 2014). In other words, approximate methods usually find near-optimal solutions, however they are time-efficient and provide a quick convergence.

4.3.1 Heuristic methods

The term ’heuristic’ stems from the Greek term “heuriskein” which is intended to mean discover or find (Martí & Reinelt, 2011). Heuristic methods are strategies designed to find good solutions to real-word problems in a reasonable time. In fact, it is not obligatory that this solution be optimal but the most important is to find it quickly, which is the primary interest of researchers, engineers and analysts (Martí & Reinelt, 2011). A heuristic method can be used within the framework of a global procedure that ensures finding an optimal solution to a problem (Martí & Reinelt, 2011). Greedy (constructive) and local search (Hill-Climbing) algorithms are popular examples of heuristic methods (Mi & Yang). Both

algorithms are widely used in CBIR (Benloucif & Boucheham, 2014; Kumari, 2015; Ohashi et al, 2003; Patil & Kokare, 2013).

Marti and Reinelt (Martí & Reinelt, 2011) expanded the classification of heuristic methods into four categories: (i) *decomposition methods*, where the original problem is divided into different sub-problems, (ii) *inductive methods*, in which, the solutions or techniques used to solve smaller or simpler cases of problems are generalized and applied to the global problem, (iii) *reduction methods*, which simplify the problem by constraining the solution space, (iv) *constructive methods* , in which a solution is built gradually from scratch (v) *local search methods*, which start with any feasible solution for the problem, then improve it progressively.

In particular, constructive and local search methods, in which the solution is constructed step by step (either from scratch or from a feasible solution) constitute the core of metaheuristic methods (Martí & Reinelt, 2011), which are described in the next section.

4.3.2 Metaheuristic methods

In contrast to the heuristics which are problem-specific methods, metaheuristics are general-purpose algorithms that can be only a bit modified to solve various optimization problems (Barrera-García et al, 2023). Metaheuristics are stochastic optimization algorithms, in which the optimization process is based on the balance between the exploration and exploitation of the search space (Barrera-García et al, 2023). Exploration and exploration are two important search mechanisms that ensure diversifying the search over the entire search space and intensifying the search in the promising regions or areas (Naik & Satapathy, 2021). A good balance between the two search mechanisms can therefore lead to good solutions with acceptable time. Metaheuristics hold today a considerable value in many scientific and technological fields such as software engineering, bioinformatics, industrial and commercial-logistics, telecommunications, economics, etc. (Nesmachnow, 2014). The great popularity of metaheuristics was in fact gained as a result of their important characteristics (Naik & Satapathy, 2021; Nesmachnow, 2014), which can be summarized in the following points:

- *Flexibility*: They are adaptable to various types of optimization problems.
- *Gradient*-free approaches: The computation of their objective functions does not need mathematical expressions like derivatives or gradients.

- *Robustness*: They are intelligent methods that can deal with imprecise, uncertain or incomplete data.

Although metaheuristics are general-purpose methods, no metaheuristic algorithm can perform with the same efficiency on every optimization problem (Naik & Satapathy, 2021). In fact, in accordance with the No-Free-Lunch (NFL) theorem (Wolpert & Macready, 1997), there is no universal optimization algorithm that can be the best for solving all classes of optimization problems. And this explains the proliferation of metaheuristic algorithms and the development of a wide variety of them. These metaheuristic algorithms are usually classified in diverse ways based on various classification criteria (Nesmachnow, 2014). The most common classification is based on the number of solutions utilized during each stage of the iterative search (Nesmachnow, 2014). According to this classification, metaheuristic algorithms are grouped into two classes: *single-solution-based metaheuristics (trajectory-based)* that operate with only one candidate solution which is improved at each search step. Examples of this class of metaheuristics are Simulated Annealing (SA) (Kirkpatrick et al, 1983), Tabu Search (TS) (Glover, 1986) and Iterated Local Search (ILS) (Lourenço et al, 2003). The second class of this classification is the *population-based metaheuristics*, in which a set of many candidate solutions (population) is used in each search step and evolved (modified and/or combined) during the iterative search, based on certain common guidelines. Examples of this class are DE, GA, PSO, Cuckoo Search, SCA, etc. Another well-known classification of metaheuristics is that based on their inspiration source (Naik & Satapathy, 2021). According to this classification, metaheuristic algorithms can be grouped into four classes (see Figure 4.1) (Ke-Lin Du, 2016; Naik & Satapathy, 2021): *evolutionary-based*, *swarm intelligence-based*, *human-based*, and *science-based algorithms*.

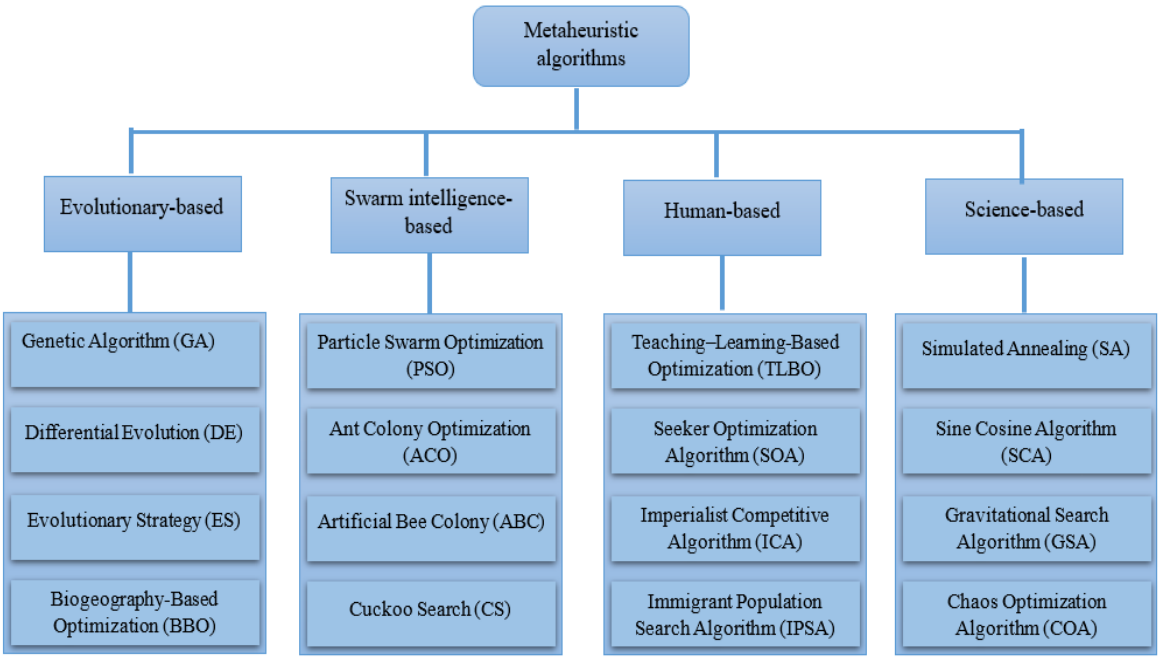


Figure 4. 1: Classification of metaheuristic algorithms based on the inspiration source

4.3.2.1 Evolutionary-based algorithms

Evolutionary Algorithms (EAs) are a class of metaheuristics that mimic the natural evolution of biological entities (Ke-Lin Du, 2016). EAs are population-based metaheuristics which can therefore explore multiple candidate solutions in parallel. They are simple to implement, easy to hybridize, and can be smoothly integrated with existing simulations or computational models for real-world problems (Ke-Lin Du, 2016).

EAs are stochastic in nature (Ke-Lin Du, 2016), they involve iterative random processes that guide the search toward the optimal or near-optimal solutions. The main components of a typical EA are a population generator and selector, a fitness function, and three genetic operators (crossover, mutation, and selection) (Ke-Lin Du, 2016). The search process of a typical EA, which is in fact inspired by the biological evolution, follows an iterative cycle that starts by initializing a random population of candidate solutions (i.e., individuals); then, a cycle of four steps, namely *evaluation*, *selection*, *variation*, and *replacement*, runs iteratively until a termination criterion (such as maximum number of generations or satisfactory solution) is met. Each iteration of the algorithm represents indeed a generation where the population can be reproduced (i.e., produce new individuals). Therefore, after initializing the first population of solutions randomly, each individual is evaluated by means of a fitness function to assess its quality. Thereafter, the best solutions

(parents) are selected to reproduce. The reproduction is in fact the generation of new individuals (offspring) using the genetic operators such the crossover and mutation. The generated offspring are subsequently evaluated using the fitness function and some of them are replaced to form a new population. The evaluation, selection, variation, and replacement steps are repeated until the termination criterion is met.

Genetic Algorithm (GA), Differential Evolution (DE), Evolutionary Strategy (ES) (Rechenberg, 1973), Evolutionary Programming (EP) (Yao et al, 1999), Genetic Programming (GP) (Koza, 1992) and Biogeography-Based Optimization (BBO) algorithm (Simon, 2008) are popular examples of EAs. Among these EAs, DE is widely used in CBIR (Boukerma et al, 2019; Gain, 2024; Grycuk et al, 2016). Thus, a detailed description of it is provided in the following.

- **Differential Evolution**

It is a widely used population-based metaheuristic that was introduced by Price and Storn in 1995 (Storn & Price, 1995). Its popularity is mainly due to its simplicity and efficiency. Indeed, Differential Evolution (DE) is simple to implement and requires only a few control parameters. In addition, it was designed to solve real-valued optimization problems (Ke-Lin Du, 2016). The selection and reproduction in DE are based on a one-to-one relationship between each parent and its offspring. A new candidate solution is created from multiple parents (multi-parent reproduction strategy) (Ke-Lin Du, 2016). Similarly to most EAs, the process of DE relies mainly on the initialization, mutation, crossover and selection steps. However, in the mutation step, the generation of new population (offspring) is carried out by perturbing current individuals with scaled differences between other individuals of the population that are randomly selected. A parent is replaced by an offspring only if the fitness function of the latter is better than that of the parent, and this which makes DE efficient and competitive.

Like most EAs, the search process of DE begins by generating randomly an initial population of N individuals, and each individual is represented by a D-dimensional real-valued vector. An individual $X_i(t)$, (i=1,…,N) at a generation t is represented as follows:

$$X_i(t)=X_{1i}(t),X_{2i}(t),...,X_{Di}(t) \tag{4.1}$$

After initializing the population, the mutation, crossover and selection steps are performed iteratively until a stopping criterion is met. In the mutation step, for each individual, a mutant vector Vi is generated by perturbing the current individual (the target vector) Xi with scaled differences of other randomly chosen individuals (vectors), whose number depends on the variant of the DE algorithm. In the DE variant referred as "DE / best / 1 / bin", the mutant vector is generated by perturbing the target vector with scaled difference of two randomly chosen vectors, as in the following equation:

$$V_i(t)=X_{r1}(t)+F\times(X_{r2}(t)-X_{r3}(t)) \tag{4.2}$$

where r_1,r_2,r_3 are distinct random indexes among the values of i (r_1,r_2,$r_3 \in \{1,2,...,N\}$), F is the scaling factor which value is a real number from the range [0,2], and t is the current generation.

In the crossover step, a new vector U_i, called trial vector, is generated from the current vector (X_i) and mutant vector (V_i) using the crossover operator as follows:

$$U_{ij}(t)=\begin{cases} V_{ij}(t) & if \quad rand(j)\leq CR \text{ or } j=rnbr(i) \\ X_{ij}(t) & otherwise \end{cases} \tag{4.3}$$

where j = 1, 2, …, D, rand(j) $\in$[0,1] is the j^{th} evaluation of a uniform random number generator, rnbr (i) is randomly selected index $\in$[1,D] and CR is the crossover rate $\in$ [0,1].

Finally, in the selection step, a comparison between the trial and the target vectors is carried out based on their fitness values (see equation (4.4)), to decide which survives to the next generation. In fact, the vector with better fitness is selected for the next generation.

$$X_i(t+1)=\begin{cases} U_i(t) & if \quad f(U_i(t))\leq f(X_i(t)) \\ X_i(t) & otherwise \end{cases} \tag{4.4}$$

DE has many variants which differ according to the adopted mutation and crossover strategies and the types of the parameters. In fact, a DE variant is generally denoted by DE/x/y/z. Where x represents the vector to be mutated, "rand" when the vector is randomly chosen, and "best" when the selected vector is the one that has the best fitness; y determines the number of difference vectors; and z indicates the crossover type, "Bin" refers to binomial and "Exp" for exponential.

4.3.2.2 Swarm intelligence-based algorithms

The optimization algorithms in this class are inspired by the collective behavior of groups of biological entities in the nature, such as birds, insects, fish, and bacteria. Swarm intelligence refers to the collective intelligence with which groups of simple agents act (Ke-Lin Du, 2016). It is specifically centered on the intelligent and cooperative behavior of swarms, which is decentralized (no central leader) and self-organized (they organize themselves naturally). Unlike EAs which improve the solutions by utilizing evolutionary operators, swarm intelligence algorithms, in contrast, create new solutions by using differential position update rules (Ke-Lin Du, 2016). Swarm intelligence methods are widely in computer vision applications as they are simple, easily adaptable to different problems and effective in solving sophisticated optimization problems (Ke-Lin Du, 2016). Among the popular swarm intelligence algorithms are PSO, which is inspired by the behavior of bird flocks, Ant Colony Optimization (ACO) (Dorigo et al, 2007), which mimics the behavior of ants in collecting foods, Artificial Bee Colony (ABC) (Basturk, 2006), which is inspired by the behavior of honey bees when searching food sources, and Cuckoo Search (CS) (Yang & Deb, 2009), which mimics the parasitic nesting behavior of cuckoo birds. In the following, we detail PSO algorithm which is widely used in CBIR (Subramanian et al, 2022; Vamsidhar et al, 2017; Vensila & Boyed Wesley, 2024; Younus et al, 2015).

- **Particle Swarm Optimization**

Particle Swarm Optimization (PSO) is a population based metaheuristic developed by Kennedy and Eberhart in 1995 (Kennedy & Eberhart, 1995). PSO is a swarm intelligence algorithm that mimics the social behavior of birds flocking or fish schooling in order to solve optimization problems. It has gained widespread popularity owing to its simplicity, lower computational cost, easy implementation (reduced code length), fast convergence and robustness of control parameters (Ke-Lin Du, 2016). In fact, the search process of PSO is inspired by the search strategy followed by birds' flock or fish school to find. More precisely, the birds (or fish) move in group toward the best food source and adjust their direction based on a collective sharing of their knowledge and experience. In the swarm moving, each member of the group (particle) adapts its position based on its own knowledge along with the knowledge acquired from its neighbors' movements, and thereby collectively adjusting the flight path. In this context, in the PSO algorithm, each candidate solution (particle) has a

velocity which is updated based on its personal best-known position (pbest) and the global best-known position found by the swarm (gbest).

Like most metaheuristics, PSO starts by creating an initial swarm (population) of N particles, each of dimension D, with random positions and random velocities. Then, the algorithm iterates by updating the particles velocity and positions until a termination criterion is reached. A particle which represents a potential solution is then defined by two variables, its current position $X_i = (x_{i,1},x_{i,2},...,x_{i,D})$ and current velocity $V_i = (v_{i,1},v_{i,2},...,v_{i,D})$. To find the optimal solution, the particles move around the search space while updating their velocities through the update of their best personal solution currently found $pbest_i = (pbest_{i,1},pbest_{i,2},\ldots,pbest_{i,D})$ and the global best solution found by swarm $gbest = (gbest_1,gbest_2,\ldots,gbest_D)$. So, after initializing the population, the fitness of each particle is evaluated and the personal and global best solutions (pbest and gbest) are updated. The position and velocity of each particle are then updated as follows:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{4.5}$$

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (pbest_i^t - X_i^t) + c_2 r_2 (gbest^t - X_i^t) \tag{4.6}$$

where ω is a constant value that represents the inertia weight; c_1 and c_2 are individual and group learning factors, which are also constant values; $r_1$and r_2 are random numbers uniformly sampled in the range [0, 1]; V_i^t and X_i^t are the velocity and current position of the particle i at iteration t; $pbest_i$ is the best position ever found by the particle i up to the iteration t; gbest is the best position ever found by the swarm up to the iteration t.

The iterative search process described above is then repeated until the stopping criterion, such as maximum number of iterations or satisfactory solution, is reached.

4.3.2.3 Human-based algorithms

This type of algorithms is inspired by several human behaviors in solving problems creatively (Ke-Lin Du, 2016). They mimic different human processes such as teaching-learning activities, social interactions, communication and decision-making, to solve complex problems. Some of the well-known human-based algorithms are Teaching–Learning-Based Optimization (TLBO) (Rao et al, 2011b), Seeker Optimization Algorithm (SOA) (Dai et al, 2006), Imperialist Competitive Algorithm (ICA) (Atashpaz-Gargari & Lucas, 2007), Immigrant Population Search Algorithm (IPSA) (Kamali et al, 2015) and Exchange Market

Algorithm (EMA) (Ghorbani & Babaei, 2014). In the following, we explain in detail the TLBO algorithm which is widely used in CBIR (Bi & Pan, 2017; Jain & Bhadauria, 2016; Sameer et al, 2021).

- **Teaching-Learning-Based Optimization**

Teaching–learning-based optimization is a population-based algorithm suggested by Rao et al. (Rao et al, 2011b) in 2011. Compared to other metaheuristics like GE and PSO, TLBO is a parameter-free algorithm (Ke-Lin Du, 2016). Indeed, it does not need specific control parameters, which makes it easy to implement. In addition, TLBO has been shown to perform better than GA, PSO, and ABC with faster convergence, hence its popularity in various fields (Ke-Lin Du, 2016). In particular, TLBO is inspired by the teaching and learning process in a classroom, where a group of learners (students) can improve their knowledge by learning from a teacher as well as from the interaction with other learners. By mimicking these two basic modes of learning, the TLBO algorithm was designed based on two main phases: *Teacher phase* and *Learner phase* (Ke-Lin Du, 2016).

The efficiency of TLBO depends on the teacher's influence on the output of learners in the class, which is reflected in their results or degrees (Ke-Lin Du, 2016; Sameer et al, 2021). The population in the TLBO algorithm consists in fact of a group of N learners, each has different designed variables which actually correspond to the different subjects provided to the learners. On the other hand, the fitness of the optimization problem in TLBO corresponds to the learners' result. To improve their results (performance), the learners get knowledge from the teacher in the teacher phase and through interaction with other learners in the learner phase. In essence, TLBO algorithm begins by initializing the population, which consists of N learners (candidate solutions), each with D subjects (designed variables). Then, the teacher and learner phases run iteratively until a stopping criterion is met.

In the first phase, which is the *Teacher phase*, the teacher, who is considered as the best solution in the population, tries to improve the performance of the class by increasing the mean value of the learners' results, which is represented by the mean value of each designed variable across all learners. After computing the learners' results mean value x_{mean}, each learner x_i updates his position according to equation (4.7).

$$x_{new,i} = x_i + r(x_{teacher} - TF x_{mean}) \quad (4.7)$$

where r is a random number ∈ [0,1] and TF is a teaching factor randomly set to either 1 or 2, whose role is to emphasize the teacher impact on the learner's average results (Ke-Lin Du, 2016).

In the second phase, which is the *Learner phase,* the learners interact mutually to enhance their knowledge. In particular, for each learner x_i, a peer learner x_j is selected randomly. Then the learner's position is updated depending on its fitness value $f(x_i)$, whether it is greater or less than the fitness value of x_j ($f(x_j)$), as described in the following equation:

$$x_{new,i} = \begin{cases} x_i + r(x_j - x_i), & f(x_j) > f(x_i) \\ x_i + r(x_i - x_j), & f(x_j) < f(x_i) \end{cases} \quad (4.8)$$

4.3.2.4 Science-based algorithms

The metaheuristics of this class are specifically inspired by the fundamental concepts and phenomena of some scientific disciplines such as physics, chemistry, mathematics and biogeography (Ke-Lin Du, 2016). In fact, they mimic physical law and rules, chemical reactions, mathematical concepts and theories, and other scientific principles to solve complex optimization problems. They therefore guide the search to the best solutions by relying on the rules and concepts derived from these scientific fields. Examples of these algorithms are Simulated Annealing (SA) (Kirkpatrick et al, 1983), Gravitational Search Algorithm (GSA) (Rashedi et al, 2009), Artificial Chemical Reaction Optimization Algorithm (ACROA) (Alatas, 2011), Chaos Optimization Algorithm (Okamoto & Hirata, 2013) and Sine Cosine Algorithm (SCA) (Mirjalili, 2016). In the following we describe sine cosine algorithm which is commonly used for feature selection, although rarely used in CBIR (Liu et al, 2025; Zivkovic et al, 2022)**.**

- **Sine Cosine Algorithm**

Sine Cosine Algorithm (SCA) is a relatively recent algorithm introduced by Mirjalili in 2016 (Mirjalili, 2016). Unlike SA, SCA is a population-based algorithm that relies on multiple candidate solutions to find the best solutions for complex optimization problems. In particular, SCA is inspired by the mathematical behavior of sine and cosine trigonometric functions to guide the search process while balancing exploration and exploitation of the search space. Since its introduction, SCA has gained significant and growing attention in

many research fields due to its simple structure, easy implementation, fast convergence, fewer parameters and strong exploration capabilities (Rizk-Allah & Hassanien, 2023). To move solutions toward the best solution, SCA used the periodicity of sine and cosine functions to balance between the global exploration and local exploitation of the search space (Liu et al, 2025). Indeed, to ensure an effective balance between these exploration and exploitation, SCA algorithm uses many random and adaptive variables to adjust and control them (Mirjalili, 2016).

Like most population-based metaheuristic algorithm SCA begins by initializing a random set of candidate solutions within the search space, then it tries to improve this set of solutions iteratively until achieving the best solutions. Thus, after generating a population of random solutions, the fitness of each solution is calculated and evaluated. The best solution found so far is maintained and considered as the destination (target) solution. The other candidate solutions update their positions relative to the best solution based on sine and cosine functions as in the following equation:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 * sin(r_2) * |r_3 P_i^t - X_i^t|, & r_4 < 0.5 \\ X_i^t + r_1 * cos(r_2) * |r_3 P_i^t - X_i^t|, & r_4 \geq 0.5 \end{cases} \quad (4.9)$$

where X^t_i represents the position of the current solution in the i-th dimension at the t-th iteration; P_i denotes the destination point (target solution) in the i-th dimension; and r_1, r_2, r_3 and r_4 are random numbers used to maintain a good balance between exploration and exploitation. In particular, the parameter r_1 is used to control and determine the movement direction of the next position area. In other words, it scales the movement of each solution toward the destination solution (inside space between the solution and the destination point) or away from the destination solution (outside space between the solution and the destination point). The parameter r_2, which is within the range [0, 2π], determines the extent to which the movement to or from the destination is directed. The parameter r_3, which is usually within the range [0, 2], is used to assign stochastic weights for the destination to either emphasize ($r_3 > 1$) or deemphasize ($r_3 < 1$) its impact on the current solution, i.e., extending the distance between them when r3 > 1 and contracting it when $r_3 < 1$. Finally, the parameter r_4, which belongs to the range [0,1], is used to choose between sine or cosine during the update of the solution position.

In addition to the metaheuristic algorithms discussed above in detail, other notable algorithms are summarized in Table 4.1, highlighting their main advantages and limitations.

Table 4.1: Summary table of some key metaheuristic algorithms

Author-Year	*Name of algorithm*	*Class (Inspiration source)*	*Control parameters*	*Advantages*	*Drawbacks*
(Holland, 1975)	Genetic Algorithm (GA)	Evolutionary-based	Probability of crossover P_c. Probability of mutation P_m. Population size.	Robustness. Global search capability. Applicable to wide range of optimization problems.	Slow convergence. Parameter sensitivity. Premature convergence.
(Kirkpatrick et al, 1983)	Simulated Annealing (SA)	Science-based	Initial temperature T_0. Cooling schedule. Cooling Rate α. Final temperature T_f.	Wide applicability. Flexibility. Global optimization ability.	Slow convergence. High computational cost. Single-solution based.
(Dorigo et al, 2007)	Ant Colony Optimization (ACO)	Swarm intelligence-based	Pheromone factor α. Heuristic factor β. Pheromone evaporation rate ρ. Pheromone deposit amount Q. Initial pheromone level τ_0. Number of ants m	Strong robustness. Parallelism (explore solutions simultaneously). High quality solutions.	High computational cost. Premature convergence. Parameter sensitivity. Difficulty in handling continuous problems
(Simon, 2008)	Biogeography-Based Optimization (BBO)	Evolutionary-based	Immigration rate λ. Emigration rate μ. Habitat suitability index HIS. Mutation Probability P_m. Keep rate. Population size.	Simplicity. Flexibility. Good global exploration ability. Computational efficiency.	Parameter sensitivity. Premature convergence. Poor convergence when dealing with complex problems.

(Yang & Deb, 2009)	Cuckoo Search (CS)	Swarm intelligence-based	Discovery probability Pa. Population size.	Less parameters. Simple and easy to implement. Efficient exploration of the search space. Strong global search capability.	Slow convergence in some applications. Limited adaptability. Less effective in complex problems.

4.4 Machine learning techniques

Machine learning (ML) is defined as a subset of artificial intelligence field, where machines can learn autonomously from experience to solve a particular problem (Schuld et al, 2015), (Udousoro, 2020). The aim of ML is to discover patterns in data (i.e., learning) to understand and make decisions on new, unknown data (Schuld et al, 2015). In particular, ML is used to help machines dealing with large amount of data to efficiently make prediction or decisions on unseen (Udousoro, 2020). Accordingly, many fields and industries have integrated ML techniques to handle the massive and increased amount of data generated every day in the modern digital world. Notably, CBIR systems have increasingly incorporated ML techniques to improve their performance (Alrahhal & Supreethi, 2024a). As mentioned above ML techniques were integrated in CBIR systems to enhance their main components and address the challenges they encounter through several aspects such as enhancing feature representations and learning effective matching measures. In fact, ML was applied in CBIR through three main approaches unsupervised, supervised and deep learning (Alrahhal & Supreethi, 2024a; Hameed et al, 2021).

4.4.1 Unsupervised learning

In the unsupervised learning, the ML algorithm is meant to discover hidden patterns in unlabeled data. In simple terms, in unsupervised learning, we ask the machine to do something without explicitly telling it how to do it (Udousoro, 2020). Unsupervised learning is particularly useful in situations where labeled data is limited or unavailable. It is also capable of adapting to new and diverse databases (Alrahhal & Supreethi, 2024b). Usually, the main tasks for which unsupervised learning is used are clustering and dimensionality reduction (Alrahhal & Supreethi, 2024a). In CBIR, clustering is a subsequent step of feature extraction, in which images are grouped based on the similarity of their corresponding

features, in such a way that the images in the same cluster are similar and each cluster represents a semantically different category of images (Alrahhal & Supreethi, 2024a; Hameed et al, 2021). K-Means clustering and its variants is one of the most frequently used unsupervised algorithms in CBIR (Alrahhal & Supreethi, 2024a; Murthy et al, 2010a; Murthy et al, 2010b; Park & Hwang, 2024). As for feature dimensionality reduction, unsupervised algorithms are used to project the high-dimensional feature space into a compact, yet robust feature space. Principal Component Analysis (PCA) (Wang et al, 2012) is recognized as a key unsupervised algorithm for feature dimensionality reduction in CBIR (Adegbola et al, 2020; Belarbi et al, 2017; Chavda & Goyani, 2020; Nguyen et al, 2009; Wadhera & Agarwal, 2022). Despite their advantages, unsupervised methods have some limitations such as the overfitting issue where the model may struggle to generalize to new and unknown data, the scalability problem which arises especially with an increasing database size, and the choice of the number of clusters which may impact the clustering performance (Hameed et al, 2021).

4.4.1.1 K-Means algorithm

K-Means is a well-known and classic example of clustering in machine learning (Schuld et al, 2015). It is termed k-Means because its working principle is based on creating k distinct clusters according to the distance between the data points, in such a way that the points with similar characteristics are grouped in the same cluster (Udousoro, 2020). In particular, a cluster is formed of the data points or feature vectors that are closest to a center point (centroid). A centroid of a cluster is the mean of all vectors that are assigned to this cluster. K-Means algorithm follows an iterative refinement process which can particularly enhance the retrieval speed of CBIR systems (Murthy et al, 2010a). To achieve an effective clustering of images and retrieve relevant images, a common way to use k-Means is to run it multiple times (Murthy et al, 2010a).

The algorithm starts by defining the number of clusters (k) and initializing the centroids of clusters by selecting randomly k vectors from the dataset as the initial centers. Then, the distances between the dataset vectors and each centroid are calculated and each vector is assigned to the nearest centroid cluster and thus the initial k clusters are formed. Subsequently, the centroids of each cluster are updated by calculating the mean of all points in the cluster. The distances between all vectors and the new centroids are then recalculated and each vector is reassigned to the cluster of the nearest centroid. This process is repeated until the centroids stabilize or a maximum number of iterations is reached.

Although its simplicity and efficiency, k-Means algorithm is sensitive to the predefined number of clusters (k) (Schuld et al, 2015). Indeed, a bad choice of k may lead to insignificant clusters. In addition, K-means is also sensitive to the initial centroid vectors and can converge to local minima, as the choice of the initial centroids could make it difficult to reach the global minima and affect overall clustering of the data (Schuld et al, 2015).

4.4.1.2 Principal Component Analysis

PCA is a statistical method typically used for dimensionality reduction. It is intended to project the data space into a low-dimensional space while preserving the important information of the data. Specifically, it employed an orthogonal transformation to convert the original data into a smaller set of new and uncorrelated variables called principal components. PCA is widely used in several fields such as computer vision, pattern recognition and signal processing, due to its key properties such as: the maximization of the feature's variance, non-correlation of the new variables, best linear approximation (by minimizing the mean-square error) and maximization of the relevant information contained in the reduced features (Adegbola et al, 2020).

The PCA process starts by computing the covariance matrix which determines the mutual variation between variables. Then, the Eigenvectors and Eigenvalues are derived from the covariance matrix. The former are vectors determining the directions in which the data vary (i.e., the principal components' directions in the feature space), and the latter are scalars that quantify the amount of variance captured in each eigenvector. Larger eigenvalue indicates that the variance in the data is more explained in the corresponding eigenvector. Subsequently, the eigenvalues are sorted in a descending order and the eigenvectors corresponding to the largest eigenvalues are selected. Hence, the most important principal components that capture the most variance (essential information) are retained. Finally, the original data is transformed and projected onto the new subspace based on the selected principal components, and hence reducing the dimension of the original data space.

4.4.2 Supervised learning

Unlike unsupervised learning, supervised learning algorithms do not have to find hidden patterns in data or grouping it autonomously because they have prior knowledge of the data labels and groups (Alrahhal & Supreethi, 2024a). That is why supervised learning algorithms are mostly used for the classification task, especially in CBIR (Alrahhal & Supreethi, 2024a;

Udousoro, 2020). In fact, a training set of labeled images is used to learn the relationship between the features and class labels of images (semantic meaning), in order to identify and predict the classes of unlabeled images in the test set. Training a supervised model on labeled databases can effectively improve the accuracy of CBIR systems and enhance the semantic retrieval in these systems (Alrahhal & Supreethi, 2024b). However, this type of machine learning algorithms is dependent on the labeled set of images and become hard time-consuming with large datasets as the labeling process is usually carried out manually by humans (Latif et al, 2019). K-Nearest Neighbors (KNN) is one of the most commonly used supervised learning methods in CBIR (Alrahhal & Supreethi, 2024b; Arunkumar & Ram, 2020; Deole & Longadge, 2014; Mardi et al, 2019).

4.4.2.1 K-Nearest Neighbors

KNN is a widely used and simple standard method for classification (Schuld et al, 2015). The fundamental assumption behind KNN is that the data points which their feature vectors are close tend to represent similar examples (Schuld et al, 2015). Here, the closeness is relative to the distance between the neighbor data points, in the sense that the nearest neighbors according to a given distance metric, such as Euclidean distance, tend to belong to the same class. As illustrated in Figure 4.2, when k=1, the new example is assigned to class 1 (blue squares) because only one nearest neighbor is considered; however, when k=3, the new example is assigned to class 2 (red triangles) which is the majority class among the three nearest neighbors. In CBIR, KNN is used to predict the class of the query image (the new data point) based on the distance between its feature vector and those of all images in the training set (Mardi et al, 2019).

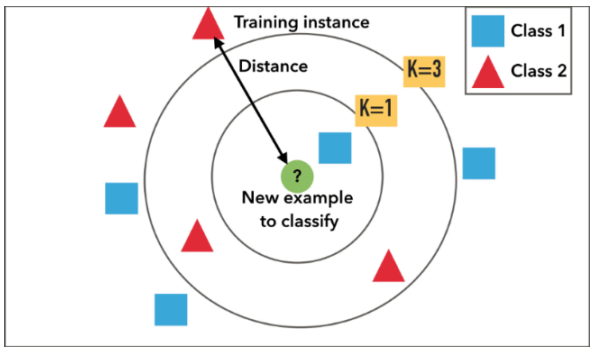


Figure 4. 2: An illustration of the K-Nearest Neighbors (KNN) algorithm (Sultani & Yousif, 2018)

The process of classification using KNN algorithm is as follows (Mardi et al, 2019):

1. Defining the number of neighbors (k): The choice of the value of the parameter k can sometimes be difficult and can greatly affect the classification performance (Schuld et al, 2015). In particular, a very large value of k may lack the locality information due to the large neighborhood; and a very small k value (limited neighborhood) may increase sensitivity to noise and produce incorrect or unreliable classification results. A common way to choose k to try different values of it (empirical testing) until finding the one that yields the best classification performance (Mardi et al, 2019).
2. Computing the distance between the new feature vector and all feature vectors in the training dataset. The common measure metric used for calculating the distance in KNN is the Euclidean distance (Mardi et al, 2019).
3. Sorting the obtained distances in an ascending order and select the k nearest neighbors (feature vectors) in the training dataset relative to the new data point (new feature vector).
4. Determining the class of the new data point by assigning it to the majority class.

4.4.3 Deep learning

Deep learning (DL) is a specific type of machine learning that has emerged as a powerful tool for dealing with massive volumes of data. DL is an advanced technology founded on the basis of Artificial Neural Networks (ANNs) (Jhaveri et al, 2022). Both ANNs and DL models are inspired by the human brain structure in which, the information is processed through progressive levels of transformation and representation (Alrahhal & Supreethi, 2024a; Jhaveri et al, 2022). However, DL models include deeper neural networks (higher number of layers), allowing them to learn more complex features such as faces, objects and emotions.

Due to their ability to model a rich variety of appearances, DL methods have achieved an impressive success in various domains, such as medical diagnostics, natural language processing, autonomous driving, text analytics, etc. (Alrahhal & Supreethi, 2024a; Jhaveri et al, 2022). Even so, the deep learning model needs to be adapted to the particular environment in which it is used to yield high-performing systems (Jhaveri et al, 2022).

Unlike traditional machine learning techniques which rely on manually extracted features for grouping or classifying data, deep learning models lean directly and optimal features from data, as long as enough training data is provided (Petscharnig et al, 2017). For

instance, deep learning models like CNNs can be used in CBIR as a feature extractor to derive high-level features directly from images (Kapadia & Paunwala, 2021). Deep learning can be used either in supervised or unsupervised learning contexts depending on the type of data (labeled or unlabeled). Among the prominent DL models used in a supervised context, CNNs are commonly utilized in CBIR (Gautam & Khanna, 2024; Kapadia & Paunwala, 2021; Ke et al, 2018). On the other hand, autoencoders are notable unsupervised deep learning models that are extensively employed in CBIR (Kale & Mukhopadhyay, 2022; Petscharnig et al, 2017; Singh et al, 2024).

4.4.3.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of deep learning that have revolutionized computer vision due to their ability to handle massive volumes of data while achieving encouraging results. The emergence of CNNs is strongly related to ANNs (Li et al, 2021c). CNNs are, in fact, a form of extension of ANNs, introducing various improvements that made them highly suitable for computer vision tasks like CBIR (Alrahhal & Supreethi, 2024a). Specifically, CNNs are more resistant to variations in scaling, rotation and translation (Alrahhal & Supreethi, 2024a). They are also able to learn more discriminative features automatically and directly from data, without needing manual feature extraction (Alrahhal & Supreethi, 2024a). This is due to their ability to incorporate deeper and larger number of layers. Nevertheless, as CNNs are supervised learning methods, they may be less effective when labeled data is not sufficient (Alrahhal & Supreethi, 2024a).

CNNs were specifically designed to process images [(Li et al, 2021c). In addition to the input layers, a typical CNN consists mainly of following types of layers (Alrahhal & Supreethi, 2024a; Li et al, 2021c):

1. **Convolution Layers:** In these layers, a sequence of filters or kernels of a specific size and number are applied on the input images to extract simple features such as edges, textures, or shapes. The output of this layer is a set of feature maps
2. **Pooling Layers**: The role of these layers is to reduce the spatial dimensions of the feature maps, allowing for decreasing the network parameters and the computational cost and increasing the translation invariance. Usually, two types of pooling are used: max pooling and average pooling

3. **Fully Connected Layers:** In this type of layers, also known as *dense layers*, each neuron is connected to all neurons of the previous layer. The output of these layers is a flattened vector (one-dimensional vector) of the purified features usable for the final prediction.

4. **Output Layer:** It is the classifier layer, in which the classification result is returned. The latter is represented by the class scores or probabilities, obtained after applying activation function such as softmax or sigmoid.

Along with these main layers, *activation layers* are essential components in the CNN architecture that produce non-linear representations by applying non-linear activation functions to the output of the previous layers. Among these functions, the ReLU function which is applied to the outputs of the convolutional layer, and the softmax or sigmoid functions which are applied to the outputs of the fully connected layers. Figure 4.3. shows an illustration of a typical CNN.

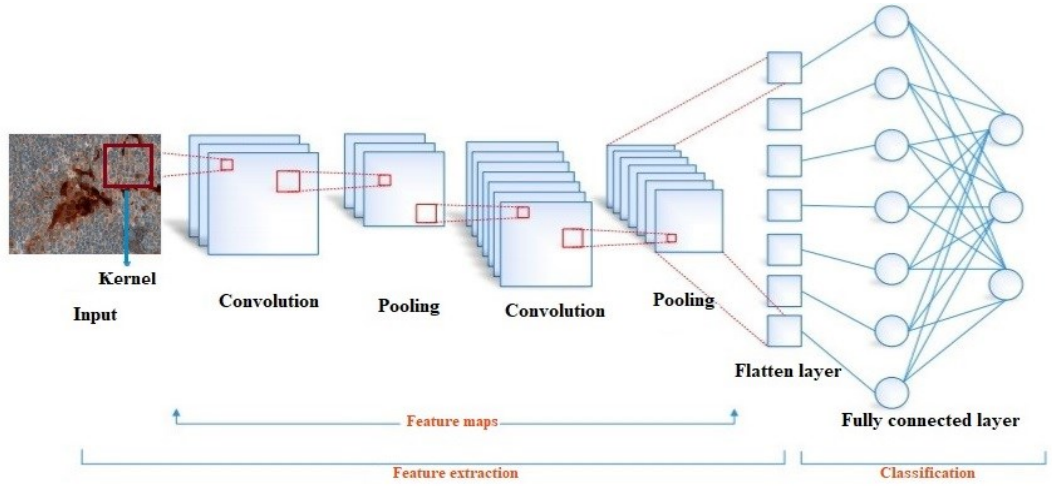


Figure 4. 3: Illustration of a typical convolutional neural network (adapted from (Hadid et al, 2023)

4.4.3.2 Autoencoders

An Autoencoder (AE) is an unsupervised neural network mainly used for dimensionality reduction (Petscharnig et al, 2017). The basic idea of AE is to learn a compact representation, called latent space, using a reconstitution process (Singh et al, 2024). This process captures the important features of the input data by reconstructing the compressed data obtained from the input data. The goal is to learn comprehensive and compact representations by making the reconstructed output as near as possible to the original input. This can be achieved by

minimizing the reconstruction loss, which is the loss between the original data and the reconstructed data (Singh et al, 2024). Overall, an AE consists of two main phases (Singh et al, 2024). In the first phase, the input data is mapped to a latent space by means of an *Encoder* network. In the second phase, the original data is reconstructed from the mapped space by means of a *Decoder network.* The loss between the original data and the reconstructed data is then measured using a loss function such as mean squared error, and minimized using an optimization algorithm such as Stochastic Gradient Descent.

Autoencoders are powerful networks that are widely used in CBIR to reduce high-dimensional image spaces to lower-dimensional image spaces, while capturing complex patterns from images and maintaining only the most discriminative features extracted from these images (Singh et al, 2024).

Table 4.2 summarizes further supervised, unsupervised and deep learning methods, complementing the representative methods described above.

Table 4.2: Summary table of additional learning methods

Author-Year	*Name of method*	*Learning approach*	*Fundamental principle*	*Advantages*	*Drawbacks*
(Fisher, 1936)	Linear Discriminant Analysis (LDA)	Supervised	Project data into a lower-dimensional space while maximizing the class separability.	Simple. Computationally efficient. Effective in scenarios with limited data.	Sensitivity to noise. High class overlap in low-dimensional Spaces. Unsuitable for complex datasets.
(Johnson, 1967; Tryon, 1939)	Hierarchical Clustering (HC)	Unsupervised	Group similar data points into clusters by creating a hierarchy of clusters represented by a tree-like diagram called dendrogram.	Simple and easy to implement. Intuitive visualization and simplification of data.	Lack of flexibility (the merge or split decisions are permanent, cannot be reversed). Sensitive to linkage criterion and distance metric.

(Cortes & Vapnik, 1995)	Support Vector Machine (SVM)	Supervised	Find the best separating hyperplane that optimally partition the data points of diverse classes so as to maximize the margin between them.	Effective with nonlinear and highly dimensional data. Good tolerance to overfitting. Superior classification performance.	Parameter sensitivity (regularization parameters, kernel type, etc.). Expensive training cost. Not suitable for large datasets.
(Bromley et al, 1993)	Siamese neural networks (SNNs)	Deep learning	Learn discriminative embeddings (mapped features) from the outputs of two (or more) identical subnetworks (CNNs) which share the same weights, then, compare the obtained embedding representations using a certain distance metric.	Less parameters to train. Lower risk of overfitting. Effective with small datasets and require fewer data to train effectively.	High computational cost. Require careful training setup and well-designed architecture. Performance depends heavily on the used distance metric for comparing the embeddings.
(Breiman et al, 1984)	Decision Trees (DT)	Supervised	Make decisions by splitting the data recursively into subsets based on feature-related question values, forming a tree-like structure that guide the decision path to the predicted class.	Simple and easy to interpret by humans. Robust against noisy and incomplete data. Exhibit high ability to learn image semantics.	Large and complicated DT may lead to wrong decisions. Less effective and difficult in handling image data (features of continues values require a discretization into categories to be usable).

4.5 Conclusion

This chapter provided a comprehensive review of optimization and machine learning techniques that have significantly advanced computer vision across different applications, including CBIR. While optimization and machine learning techniques encompass a wide range of algorithms and models, this review selectively addressed those that are most commonly used and impactful within CBIR, which is the central focus of this thesis. The chapter highlighted the role of optimization and ML techniques in enhancing the key

components of CBIR systems, and how they were employed for addressing the challenges facing these systems. We provided a structured taxonomy of optimization and machine learning techniques, including heuristic and metaheuristic methods for optimization, and supervised, unsupervised and deep learning methods for machine learning. For each category, we reviewed the algorithms commonly used in CBIR.

After reviewing the literature related to CBIR as well as the impactful optimization and machine learning techniques within it, the next chapters present our proposed contributions that attempt to address the main challenges encountered by CBIR systems, while leveraging the potential of optimization and ML techniques.

Part III

Personal Contributions

Chapter 5

Developing Discriminative and Robust Feature Representations: New methods for effective CBIR

5.1 Introduction

In the previous chapters of this thesis, we presented a comprehensive review of content-based image retrieval existing methods, as well as the common optimization and machine learning techniques used to address its key issues. Building upon the presented review, existing methods in CBIR continue to suffer from the gap between the machine low-level description of images and their corresponding high-level semantics. This fundamental issue highlights the need for more effective feature representations able to reduce the semantic gap and thereby enhance the retrieval performance. Motivated by this requirement, we introduced two contributions that are presented in this chapter. The first contribution consists of a new color-texture descriptor named "O-OCLBP" (Boukerma et al, 2025a), which builds on the existing OCLBP method (Mäenpää & Pietikäinen, 2004) while using a novel inter-channel computation scheme to enhance its performance. We have farther applied the proposed scheme to another variant of OCLBP, which is IOCLBP, enabling the construction of another robust descriptor that we have termed "O-IOCLBP". The two proposed descriptors enhance the robustness of the original descriptors to illumination change and significantly improve the retrieval performance. The second contribution is an enhanced feature representation obtained by applying a dynamic weighting scheme on four local pattern methods. The proposed approach named "DPW" (Boukerma et al, 2022) utilizes differential evolution algorithm to generate an optimal weight vector for each class of the image database. These weight vectors serve to weight dynamically the patterns extracted from images, thereby minimizing the distance between images and improving the retrieval results. In the following sections a detailed description of the two contributions is provided, including the background

context and a review of related work, an in-depth description of the proposed methodology, and an evaluation based on experimental results.

5.2 O-OCLBP and O-IOCLBP: new color-texture descriptors

5.2.1 Overview

Color is recognized as the most used visual characteristic of images, owing to the discriminative power it has gained from the capacity of the human eye to distinct its different values. Furthermore, it has also been noted that texture is an important and commonly used visual characteristic due to the straightforward perception of its different attributes, such as smoothness, roughness and regularity of patterns, by the human eye. In particular, it has been highlighted in chapter 3 that local binary pattern method and its variants have gained significant attention in CBIR research. Typically, these methods are recognized for their discriminative power and computational simplicity. However, the original LBP and its grayscale variants exhibit several limitations, especially their sensitivity to noise and less robustness caused by their lack of rotation and illumination invariance. On the other hand, it was reported that incorporating the color information with the texture information captured using LBP can significantly improve its effectiveness and discrimination capability (Agarwal et al, 2019; Bianconi et al, 2017).

Opponent color local binary patterns (OCLBP) (Mäenpää & Pietikäinen, 2004) is one of the earliest LBP extensions that has incorporated both color and texture image characteristics in a single descriptor. This descriptor, which was originally developed for the classification task, has enhanced the discriminative power of LBP and its robustness to illumination changes by incorporating the color information from an opponent color space. Specifically, OCLBP combines several LBP histograms computed from each color channel independently to constitute the intra-channel features on the one hand, and on the other hand the LBP histograms computed on couples of color channels jointly to form the inter-channel features. Improved opponent color local binary patterns (IOCLBP) is an extended version of OCLBP that has improved its invariance to illumination changes by substituting the point-to-point thresholding used in OCLBP with a point-to-average thresholding. For more details about OCLBP and IOCLBP, see chapter 3.

In order to enhance the OCLBP performance and increase its invariance to illumination changes, we introduce a novel scheme for the computation the inter-channel features of OCLBP, while using the same computation method employed in OCLBP to derive the intra-channel features. Our proposed descriptor named orthogonal OCLBP (O-OCLBP) is thus constructed by the combination of the intra-channel features and the new inter-channel features. The latter are derived by applying the LBP operator on couples of color channels jointly in an orthogonal topology instead of the circular topology used in OCLBP. Moreover, to enhance the Improved OCLBP (IOCLBP) performance, we apply our proposed scheme to compute the IOCLBP's inter-channel features, which are combined with the intra-channel features, computed as in IOCLBP, to derive another descriptor named orthogonal IOCLBP (O-IOCLBP).

An in-depth description of the proposed methodology, the experimental setup and results and results is provided in the following sections.

5.2.2 Proposed methodology

In the OCLBP method, the final feature vector is obtained by the concatenation of the intra- and inter-channel features. Specifically, to derive the intra-channel features, the LBP operator is applied on each channel of the color space. As for the inter-channel features, the LBP operator is applied on the pairs of color channels to derive six color LBP features, from which only three are used as only one of each pair of the opposing channel pairs is retained. See Figure 3.6 for further clarification of the computation of the original OCLBP in the RGB color space. As depicted in the figure, in the computation of the inter-channel features, the distribution of the pairs of color channels is performed in a circular topology, in a manner such that the neighbor pixels are taken from a color channel and the center pixel is taken from another channel. To enhance the invariance of OCLBP to illumination change and capture more lighting variation, we propose to use another scheme for distributing the pairs of color channels when computing the inter-channel features. In essence, our idea is to introduce a novel distribution of the pairs of color channels when forming the inter-channel features. We then incorporate the two colors of each pair of channels in the neighboring itself, and distributed these two colors between the neighbor pixels in an orthogonal topology instead of taking all the neighbor pixels from the same color channel and the center pixel from the second color. More precisely, the opponent color channels are distributed according to three

directions: horizontal, vertical, and diagonal. In particular, the horizontal and vertical symmetric neighbor pixels are picked from the same color channel of the center pixel, and the diagonal symmetric neighbor pixels are picked from another color channel. Figure 5.1 illustrates an example of the computation of our proposed orthogonal OCLBP in the RGB color space (Boukerma et al, 2025a). For instance, for the RG pair of color channels, the center pixel is picked from the red channel; while, as distinct from OCLBP, the neighbor pixels are not all taken from the green channel. In fact, the horizontal and vertical neighbor pixels are taken from the red channel, and the diagonal neighbors are taken from the green channel. Similarly, the O-OCLBP codes for the RG and GB couple of color channels are derived (see Figure 5.2). The final O-OCLBP feature vector is then obtained by the combination of the three intra-channel LBP histograms, which are derived as in OCLBP, and the three inter-channel features derived as previously explained.

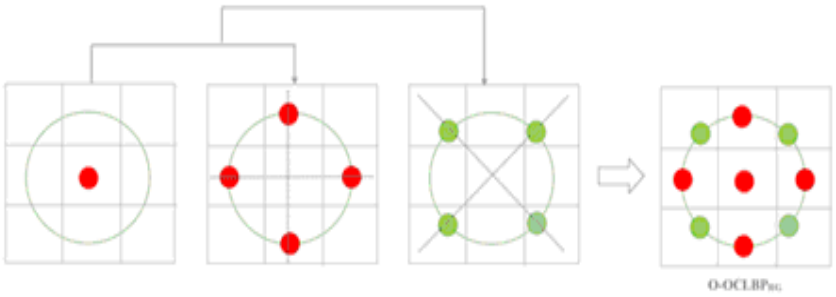


Figure 5. 1: Sampling scheme of the construction of the O-OCLBP operator for the RG pair of color channels with an 8-pixel neighborhood

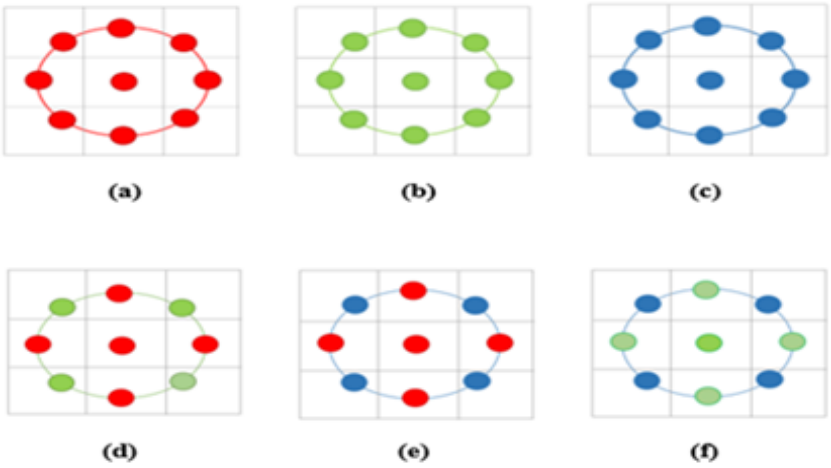


Figure 5. 2: Illustration of the O-OCLBP code construction in the RGB color space, for: (a) the RR pair, (b) GG pair, (c) BB pair, (d) RG pair, (e) RB pair, (f) GB pair (Boukerma et al, 2025a)

This effective yet simple scheme is also applied in the same way to the improved version of OCLBP (IOCLBP) to derive the O-IOCLBP features formed by the concatenation of the intra-channel features and the modified inter-channel features. Both proposed methods are further explained in the following sections.

— O-OCLBP computation

To compute the proposed O-OCLBP operator, the relationship between the center pixel and the orthogonally symmetric neighboring is encoded .This implies that the LBP descriptor is applied on each pair of color channels by considering the horizontal, vertical and diagonal directions.

For a 3×3 window, in which 8 neighbor pixels are surrounding the center pixel and arranged as shown in Figure 5.3, the O-OCLBP operator is computed as in equation (5.1). The four horizontal and vertical neighbor pixels are then thresholded by their center pixel value while picking them from the same color channel. On the other hand, the four diagonal neighbor pixels are thresholded by the center pixel value while picking them from a distinct color channel (i.e., if the center pixel is picked from the first color of the channel pair, the diagonal neighbors are taken from the second color channel).

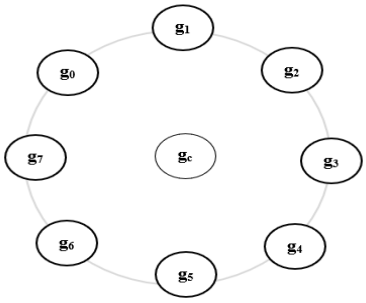


Figure 5. 3: Structure of the 8 neighbors of a pixel

Mathematically, for a 3×3 block image of P neighbors (P = 8), lying around the center pixel c, and a pair of channels (M, N), the O-OCLBP code of a given pixel c is defined as follows:

$$O\text{-}OCLBP(x_{c,M}, y_{c,M}) = \sum_{p=0}^{(P/2)-1} S(g_{2p,N} - g_{c,M})2^{2p} + \sum_{p=0}^{(P/2)-1} S(g_{2p+1,M} - g_{c,M})2^{2p+1} \quad (5.1)$$

where P is the number of neighboring pixels (P = p_0, p_1, …, p_7); $g_{2p,N}$ is the intensity of the (2p)-th pixel in the channel N (in the diagonal neighborhood); $g_{2p+1,M}$ is the intensity of the (2p + 1)-th pixel in the channel M (in horizontal or vertical neighborhood); $g_{c,M}$ is the center pixel intensity; and finally, the binary threshold function S is defined as in equation (3.2).

To form the final O-OCLBP histogram, the O-OCLBP codes for all color channel combinations are first computed; then, the resulting six histograms (representing the three intra-channel features and the three inter-channels features) are concatenated.

— O-IOCLBP computation

To improve the robustness of the improved OCLBP (IOCLBP), the proposed orthogonal scheme applied on OCLBP to derive the inter-channel features is further applied to IOCLBP. Similarly, to O-OCLBP, the O-IOCLBP features are obtained by the concatenation of the intra- and inter-channel features. The former are derived in the same way as in IOCLBP; however, the latter are derived by applying the orthogonal scheme. In particular, to the inter-channel features are extracted so that the horizontal and vertical symmetric neighbor pixels are picked from the same color channel of the center pixel, while the diagonal symmetric neighbor pixels are picked from the second color channel. The difference here, between O-IOCLBP and O-OCLBP lies in the encoding scheme of both intra-and inter-channel features. Specifically, in O-IOCLBP, the mean (m_M) of the local neighborhood is used as a threshold to compare the neighboring pixels instead of using the center pixel intensity as a threshold such as in O-OCLBP. The mean (m_M) is defined in equation (5.3). The O-IOCLBP code is therefore computed as in equation (5.2).

For a pair of channels (M, N) and a 3×3 image window, the O-IOCLBP code of a given pixel c with P neighbors (P = p_0, p_1, …, p_7) is mathematically computed as follows:

$$O\text{-}IOCLBP(x_{c,M}, y_{c,M}) = \left(\sum_{p=0}^{(P/2)-1} S(g_{2p,N} - m_M)2^{2p} + \sum_{p=0}^{(P/2)-1} S(g_{2p+1,M} - m_M)2^{2p+1}\right) + S(g_{c,M} - m_M)2^P \quad (5.2)$$

$$m_M = \frac{g_{c,M} + \sum_{P=0}^{P-1} g_{p,M}}{P+1} \quad (5.3)$$

where m_M is the mean of all pixels in the block, knowing that these pixels are considered in the channel M; $g_{c,M}$ is the intensity of the center pixel in the channel M; $g_{2p,N}$ is the intensity of the (2p)-th pixel in the channel N (representing the neighbor pixels in the diagonal direction); $g_{2p+1,M}$ is the intensity of the (2p+1)-th pixel in the channel M (representing the neighbor pixels in the horizontal or vertical direction) ; and the function S is defined as in equation (3.2).

Practically speaking, for a neighborhood of 8 pixels, the dimension of the final O-OCLBP is 1536 bins and that of O-IOCLBP is 3069 bins (1533 for the intra-channel features and 1536 for the inter-channel features).

5.2.3 Experimental evaluation

In order to retrieve relevant images, we extract color texture features using the two proposed descriptors, namely O-OCLBP and O-IOCLBP. To assess the effectiveness of the proposed descriptors, they are compared to some state-of-the-art descriptors using different several image datasets. In particular, the proposed O-OCLBP and O-IOCLBP are compared to the baseline methods (OCLBP and IOCLBP), as well as to color LBP and color ILBP methods. The color LBP features are derived by applying the basic LBP operator on each color channel of the color space, then concatenating the resulting three histograms corresponding to each color channel to form the final color LBP feature vector. Similarly, the ILBP operator is applied on the three components of the color channel to derive the color ILBP feature vector. In fact, we primarily opt for RGB color space to compute all the color texture features mentioned previously; then, we expand the computation to further color spaces to conduct a more thorough comparison. These additional color spaces are in fact HSV, CIELAB and YCbCr.

5.2.3.1 Experimental details

In this section, we describe the experimental framework used to carry out the different experiments of this work, including the experimental setup, the used image datasets, the performance evaluation and matching measure metric employed.

- **Experimental setup:** This approach was implemented using MATLAB language, on an Intel(R) Core i7, 2.6 GHz CPU, 8 GB internal RAM and 64-bit Windows 10 operating system.

- **Image datasets:** To evaluate the performance of the proposed descriptors, a variety of image datasets was used, including three general-purpose datasets (Corel 1-K, Corel-10K and COIL-100) and five color textural image datasets (KTH, KTH-TIPS2b, STex, USPTex and NewBarkTex). Details and samples from each dataset can be found in Chapter 2 (section 2.7).

- **Performance evaluation:** To assess the performance of the proposed methods, the average precision rate and average recall rate (APR and ARR) metrics are used. The retrieval of images in these experiments was performed for the 10 images returned. Additionally, it should be noted that all images in the image database were considered as query images.

- **Matching measures:** To ensure the robustness and effectiveness of the proposed methods, a fair and thoroughly comparison of their performance across various distance metrics is performed. These distance metrics are: Euclidean, Manhattan, Canberra, Chebyshev and Square chord distance. For more details of these metrics and their formulas, refer to Chapter 2 (section 2.5).

It is worth nothing that in most experiments, the features were extracted in the RGB color space. Furthermore, additional experiments were performed so that the features were extracted in HSV, CIELAB and YCbCr color spaces. As well, the Manhattan distance was basically used in most experiments to compare images. In addition, Euclidean, Canberra, Chebyshev and Square chord distances were used for evaluating the proposed descriptors in further experiments.

5.2.3.2 Results and analysis

This section presents and analyzes the results obtained from the different experiments described in the previous section. In all subsequent tables, the best results are highlighted in bold.

A. Comparison with the baseline methods

To demonstrate improvements achieved by our proposed methods over the local pattern methods related to them, the proposed O-OCLBP or O-IOCLBP are compared to the color LBP, color ILBP, OCLBP and IOCLBP methods. In particular, the proposed O-OCLBP is compared to the color LBP and OCLBP, and the proposed O-IOCLBP is compared to the

color ILBP and IOCLBP. In this experiment, all the local pattern features were coded in the RGB color space.

Table 5.1 presents a comparative analysis of the performance between the proposed and baseline methods using five different distance metrics. In this experiment, the APR obtained for the top 10 retrieved images and the ARR for the top 100 retrieved images using the different methods are compared.

Table 5.1: Performance comparison using various distance metrics on Corel-1k dataset (Boukerma et al, 2025a)

Distances	*Manhattan*		*Euclidean*		*Canberra*		*Chebyshev*		*Square chord*	
Methods	*APR*	*ARR*	*APR*	*ARR*	*APR*	*ARR*	*APR*	*ARR*	*APR*	*ARR*
Color LBP	69.36	42.46	62.09	34.92	69.94	39.53	49.63	29.19	70.29	43.11
OCLBP	74.43	48.68	61.32	40.01	74.28	42.06	56.97	**36.88**	75.22	48.83
Proposed O-OCLBP	**76.24**	**49.66**	**67.82**	**41.85**	**77.48**	**46.10**	**57.65**	35.29	**76.21**	**49.45**
Color ILBP	71.65	44.76	64.00	37.08	71.89	39.96	52.73	30.43	71.01	44.07
IOCLBP	77.12	51.16	64.41	42.14	73.77	41.71	**59.04**	**38.58**	76.46	50.63
Proposed O-IOCLBP	**78.65**	**52.29**	**70.27**	**43.79**	**77.40**	**44.57**	58.80	35.38	**77.78**	**51.23**

As observed in Table 5.1, it is evident that the proposed O-OCLBP achieves the best performance (either in terms of APR or ARR), compared to OCLBP and color LBP, for the majority of distances. On the other hand, the performance of the proposed O-IOCLBP is better than that of IOCLBP and color ILBP in most cases. More precisely, the best APR achieved by O-OCLBP is obtained using Canberra distance (77.48%), with a margin of 3.2% relative to OCLBP. In addition, the best ARR achieved by O-OCLBP is obtained using the Manhattan distance (49.66%), with a margin of 0.98% relative to OCLBP. As for the proposed O-IOCLBP, the best APR and best ARR are obtained with the Manhattan distance (best APR=78.65% and best ARR=52.29%), with a margin of 1.53% and 1.13% respectively, compared to IOCLBP. It is also noteworthy that the two proposed descriptors particularly showed a remarkable improvement in terms of APR when using the Euclidean distance. In fact, the APR of the proposed O-OCLBP and proposed O-IOCLBP were increased to 67.82% (with a gain of 6.5%) and to 70.27% (with a gain of 5.86%) compared to OCLBP and IOCLBP, respectively.

Tables 5.2 and 5.3 present a comparison of the retrieval performance in terms of APR and ARR, both for the top 10 retrieved images. The local pattern features were computed in

the RGB color space over all the image databases mentioned above, namely Corel-1K, Corel-10K, COIL-100, KTH-TIPS, KTH-TIPS2b, NewBarkTex, Stex and USPTex.

Table 5.2: Average precision rates computed in RGB color space using Manhattan distance (Boukerma et al, 2025a)

Datasets / *Methods*	*Corel-1K*	*Corel-10K*	*COIL-100*	*KTH-TIPS*	*KTH-TIPS2b*	*New-BarkTex*	*Stex*	*USP-Tex*
Color LBP	69.36	42.20	78.42	84.00	92.95	80.12	72.26	72.24
OCLBP	74.43	43.76	82.47	88.28	95.98	80.24	79.53	78.96
Proposed O-OCLBP	**76.24**	**47.04**	**83.46**	**89.05**	**96.26**	**82.18**	**81.25**	**80.21**
Color ILBP	71.65	44.72	78.95	85.74	94.13	77.70	80.83	77.63
IOCLBP	77.12	46.87	83.19	87.89	**96.60**	79.85	82.53	81.88
Proposed O-IOCLBP	**78.65**	**49.92**	**83.74**	**88.72**	96.50	**81.46**	**83.42**	**82.51**

As shown in Table 5.2, the proposed O-OCLBP descriptor increased the APR compared to the baseline descriptors by a significant margin, in all cases. Specifically, a marked improvement is observed with Corel-10K dataset. In fact, the APR with the latter achieves 47.04% with gain of 3.28% over OCLBP. On the other hand, the proposed O-IOCLBP increased the APR of almost all image databases, except for KTH-TIPS2b. The meaningful improvement achieved by O-IOCLBP was also achieved on Corel-10K, with an APR of 49.92% yielding a 3,05 % gain over IOCLBP. Overall, when comparing the average precision obtained over all databases, it is noteworthy that the two proposed descriptors achieve the best APR with the KTH-TIPS2b database.

Table 5.3: Average recall rates computed in RGB color space using Manhattan distance (Boukerma et al, 2025a)

Datasets / *Methods*	*Corel-1K*	*Corel-10K*	*COIL-100*	*KTH-TIPS*	*KTH-TIPS2b*	*New-BarkTex*	*Stex*	*USP-Tex*
Color LBP	6.94	4.22	7.84	10.37	2.15	2.94	45.16	60.20
OCLBP	7.44	4.37	8.25	10.90	2.22	2.95	49.70	65.80
Proposed O-OCLBP	**7.62**	**4.70**	**8.34**	**10.99**	**2.23**	**3.02**	**50.78**	**66.84**
Color ILBP	7.16	4.47	7.89	10.58	2.18	2.86	47.83	64.69
IOCLBP	7.71	4.69	8.32	10.85	**2.24**	2.93	51.58	68.24
Proposed O-IOCLBP	**7.86**	**4.99**	**8.37**	**10.95**	2.23	**2.99**	**52.14**	**68.76**

As for the average recall results which are presented in Table 5.3, it is evident that the proposed O-OCLBP and O-IOCLBP outperform the color ILBP and IOCLBP on almost all image databases, except KTH-TIPS2b when IOCLBP outperforms O-IOCLBP. Overall, the ARRs obtained over all image databases reveal that the best results were obtained with USPTex database, with an ARR of 66.84% when using O-OCLBP and 68.76% when using O-IOCLBP.

Further comparison in regards to color spaces is provided in Tables 5.4 and 5.5. In particular, a comparison between the proposed and baseline descriptors is carried out in various color spaces, namely RGB, HSV, CIELAB and YCbCr, on one general-purpose image database (Corel-1K) and one color textural image database (KTH-TIPS2b). In fact, the APR and ARR results for both Corel-1K and KTH-TIPS2b image databases were computed for the top 10 retrieved images.

Table 5.4: Average precision rates computed in different color spaces on Corel-1K and KTH-TIPS2b databases, using Manhattan distance (Boukerma et al, 2025a)

Color spaces / *Methods*	*COREL-1K*				*KTH-TIPS2b*			
	RGB	*HSV*	*CIE-Lab*	*YCbCr*	*RGB*	*HSV*	*CIE-Lab*	*YCbCr*
Color LBP	69.36	74.12	71.00	71.57	92.95	94.78	95.39	96.36
OCLBP	74.43	72.04	71.28	69.20	95.98	94.79	**95.91**	94.48
Proposed O-OCLBP	**76.24**	**75.53**	**73.70**	**73.84**	**96.26**	**95.64**	95.58	**96.47**
Color ILBP	71.65	**59.68**	74.71	74.09	94.13	81.11	**96.71**	**96.87**
IOCLBP	77.12	57.68	72.44	72.73	**96.60**	85.52	96.19	95.93
Proposed O-IOCLBP	**78.65**	59.19	**76.02**	**77.21**	96.50	**85.87**	96.46	96.76

Table 5.5: Average precision rates computed in different color spaces on Corel-1K and KTH-TIPS2b databases, using Manhattan distance (Boukerma et al, 2025a)

Color spaces / *Methods*	*COREL-1K*				*KTH-TIPS2b*			
	RGB	*HSV*	*CIE-Lab*	*YCbCr*	*RGB*	*HSV*	*CIE-Lab*	*YCbCr*
Color LBP	6.94	7.41	7.10	7.16	2.15	2.19	2.21	2.23
OCLBP	7.44	7.20	7.13	6.92	2.22	2.19	**2.22**	2.19
Proposed O-OCLBP	**7.62**	**7.55**	**7.37**	**7.38**	**2.23**	**2.21**	2.21	**2.23**

Color ILBP	7.16	**5.97**	7.47	7.41	2.18	1.88	**2.24**	**2.24**
IOCLBP	7.71	5.77	7.24	7.27	**2.24**	1.98	2.23	2.22
Proposed O-IOCLBP	**7.86**	5.92	**7.60**	**7.72**	2.23	**1.99**	2.23	2.24

The results on the two tables clearly indicate that, on *Corel-1K* database, the proposed O-OCLBP achieves the best results in terms of both APR and ARR, in all color spaces. On the other hand, the proposed O-IOCLBP reaches the best results in three color spaces, namely RGB, CIELAB and YCbCr, for both APR and ARR. However, in the HSV color space, although O-IOCLBP performs better than IOCLBP, its performance remains slightly less than ILBP for both APR and ARR, with 59.19% against 59.68% and 5.92% against 5.97%, respectively. As for *KTH-TIPS2b* database, the proposed O-OCLBP method exhibit better APR results in the RGB, HSV and YCbCr color spaces, with 96.26%, 95.64%, and 96.47%, respectively. However, the proposed O-IOCLBP performs better only in the HSV color space with an APR of 85.87% and ARR of 1.99%. However, it is readily apparent that even when the baseline methods outperform the proposed O-IOCLBP, the difference is marginal.

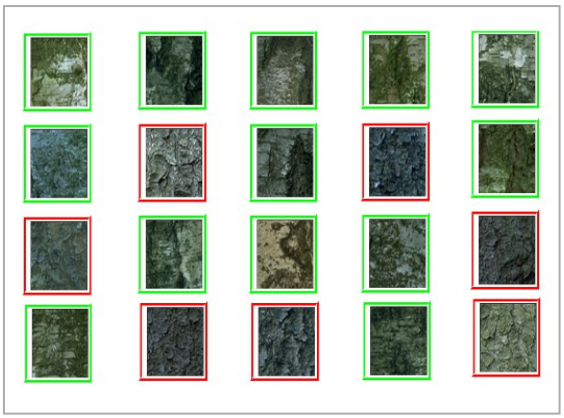
Two examples of query images from Corel-1K and NewBarkTex databases are illustrated in Figures 5.4 and 5.5, respectively. In this experiment, 20 images from Corel-1K database were retrieved using color LBP, OCLBP and O-OCLBP, respectively (Figure 5.4 (a), (b) and (c)). In addition, 20 images were also retrieved from NewBarkTex using color ILBP, IOCLBP and O-IOCLBP, respectively (Figure 5.5 (a), (b) and (c)). Knowing that, the features were computed in the RGB color space using the Manhattan distance.

(a)

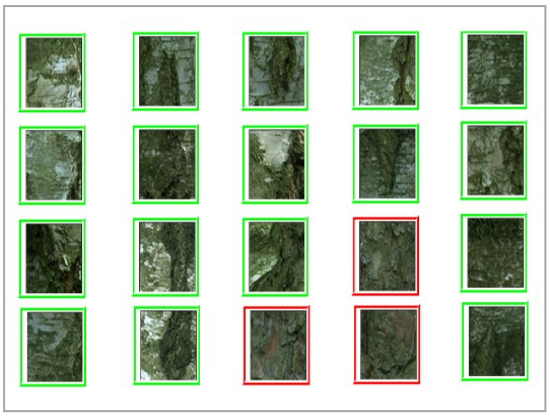
(b)

(c)

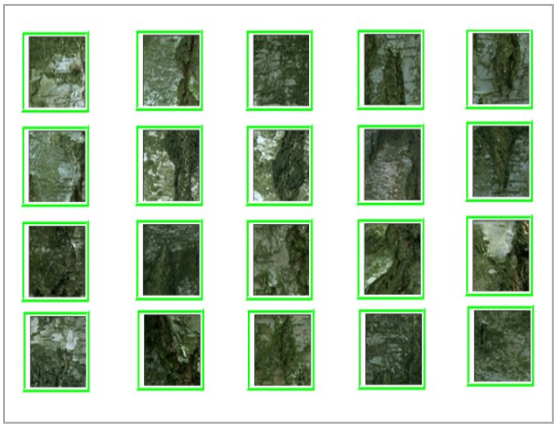
Figure 5. 4: Example of a query image from Corel-1K database using: (a) color LBP, (b) OCLBP and (c) O-OCLBP (Boukerma et al, 2025a)



(a)



(b)



(c)

Figure 5. 5: Example of a query image from NewBarkTex database using: (a) color ILBP, (b) IOCLBP and (c) O-IOCLBP (Boukerma et al, 2025a)

From Figure 5.4, it can be observed that the precision obtained when using the color LBP was 13/20, then it was increased to 15/20 with OCLBP and 18/20 with the proposed O-

OCLBP. For the second query image which is illustrated in Figure 5.5, the precisions were 13/20, 17/20 when using the color ILBP and IOCLBP, respectively; and reached 20/20 when using the proposed O-IOCLBP.

B. Comparison with other state-of-the-art methods

In addition to the baseline methods, the two proposed descriptors are also compared to other state-of-the-art descriptors developed basically for image retrieval. These descriptors are: maLBP, mdLBP, Edgy Salient LBP+ 64 bin quantized RGB (Galshetwar et al, 2017), MLCDMH (Pradhan et al, 2019), LDOP (Dubey & Mukherjee, 2020) and MMLBP descriptors.

Table 5.6 provides a comparison of the performance between the two proposed descriptors and the above-mentioned state-of-art descriptors on Corel-1K database. Specifically, the presented APR results were computed for the top 10 retrieved images, and ARR results were obtained for the top 100 retrieved images. It should be noted that performances of Edgy Salient LBP+ 64 bin quantized RGB descriptor and LDOP are reported from (Vimina & Divya, 2020).

Table 5.6: Comparison of the performance between the proposed descriptors and some state-of-art methods on Corel-1K database (Boukerma et al, 2025a)

Methods	*Average Precision Rate (APR)*	*Average Recall Rate (ARR)*
maLBP	71.86	-
mdLBP	73.63	-
Edgy Salient LBP+ 64 bin quantized RGB descriptor	73.82	45.00
MLCDMH	64.00	-
LDOP	60.21	37.7
MMLBP	77.92	48.57
Proposed O-OCLBP	76.24	49.66
Proposed O-IOCLBP	**78.65**	**52.29**

The obtained APR and ARR results reveal that our proposed O-OCLBP descriptor outperforms almost all the state-of-the-art descriptors. Moreover, the proposed O-IOCLBP demonstrates a superiority over all the evaluated state-of-the-art descriptor, with an APR of 78.65% and an ARR of 52.29%.

C. Evaluation of the proposed methods in presence of noise

A challenging test for our proposed O-OCLBP and O-IOCLBP descriptors is to use them to derive relevant features in a noisy environment. To evaluate the impact of noise on the performance of the proposed descriptors, we added noise to all query images of the defined database. To simulate the noise effect, we contaminated all the query images by "*salt and pepper*" noise (which is a widely used type of impulse noise), while maintaining clear images in the database.

Table 5.7 provides a comparison of the performance between the proposed methods (without and with noise) and some the state-of-the-art methods, including OCLBP, IOCLBP, mdLBP and MMLBP. More precisely, the APRs and ARRs for all methods were first computed for clear query images (without applying noise); then, "*salt and pepper*" noise with a density of 0.1 was applied proposed methods (noisy query images). In this experiment, all features were computed in RGB color space, using the Manhattan distance. The ARP and ARR results were obtained for the top 100 images retrieved from Corel-1K and NewBarkTe*x* databases.

Table 5.7: Comparison of the performance between the proposed descriptors (in both absence and presence of noise) and some state-of-art-descriptors for the top 100 images (Boukerma et al, 2025a)

	COREL-1K		*NewBarkTex*	
Methods	*APR*	*ARR*	*APR*	*ARR*
OCLBP	48.68	48.68	55.86	20.54
mdLBP	-	44.07	-	
IOCLBP	51.16	51.16	54.79	20.14
MMLBP	-	48.57	-	-
Proposed O-OCLBP	49.66	49.66	**57.32**	**21.07**
Proposed O-OCLBP (in presence of noise)	48.70	48.70	55.90	20.55
Proposed O-IOCLBP	**52.29**	**52.29**	56.98	20.95
Proposed O-IOCLBP (in presence of noise)	51.63	51.63	56.60	20.81

From Table 5.7, it can be observed that in presence of noise the APR and ARR results of the proposed methods were decreased, however, they are comparatively high as opposed to the other state-of-the-art methods. In fact, the performance of the proposed O-OCLBP and O-IOCLBP computed for noisy images are still considerably high compared to the state-of-the-art methods which were computed for clear images. Here, there is a clear indication that the proposed descriptors exhibit a certain level of noise resistance.

5.3 Dynamic pattern weighting approach for enhancing feature representations

5.3.1 Overview

As outlined in the previous chapters, feature weighting is an important technique that can contribute effectively to enhance the representation of images by assigning different degrees of importance to the features representing images. In particular, FW techniques were successfully applied to local pattern methods to enhance their performance. In fact, there are many LBP-based weighting approaches, which have been deeply explored in Chapter 3 (see section 3.6.2). Among these approaches is the pattern weighting approach, in which the weights are assigned to the individual patterns extracted through the LBP method (or its variants). An example of the works that have used this approach is the work of (Boukerma et al, 2019), in which a local pattern weighting framework based on DE algorithm was proposed to enhance the CBIR performance. The DE algorithm was used to generate an optimal weight vector whose values were associated with the patterns constituting the LBP histogram. This weighting framework was also applied to other LBP variants histograms, namely ILBP, LTP and MLBP. To enhance the weighting framework introduced in the work of (Boukerma et al, 2019), we propose in this study, a weighting mechanism inspired by that proposed in (Boukerma et al, 2019), but instead of generating a single weight vector, many weight vectors are generated to weight the local patterns. The number of these weight vectors is equal to the number of classes in the image dataset.

Thus, in our dynamic pattern weighting (DPW) mechanism, instead of generating a global weight vector for the entire image dataset, which cannot be adapted to all types of images in the dataset, different weight vectors are generated and associated dynamically to each query image based on the assumed class to which the query image belongs. To find the predicted class of the query image, we use a supervised learning algorithm to classify the query image. Specifically, we use KNN algorithm to predict the class of the query image; then, we assign the weight vector generated for this class to the patterns extracted from the query image. Furthermore, to make the selected weight vector more adapted to the query image, its values are further refined and adjusted according to the importance of the pattern in the image. More details of the proposed approach and the obtained results are discussed in the following sections.

5.3.2 Proposed methodology

Overall, the proposed CBIR system relies on three main phases: feature extraction, optimization and retrieval. The general architecture of the proposed CBIR system is illustrated in Figure 5.6.

—F**eature extraction:** This phase is first preceded by a pre-processing step which consists in converting all images in the database into a gray level. Knowing that the image database is divided into two sets: the test database which contain 80% of the database images, and an optimization database which contain the remaining 20% of images. After the pre-processing step, the features are extracted from images using one of the above-mentioned local pattern descriptors. In particular, we first use the basic LBP method to extract the features, then we utilize the other LBP variants — ILBP, LTP and MLBP— to confirm the effectiveness of the proposed weighting technique.

Considering a 3×3 window, with 8 neighbors and a radius equal to 1, we then extract local pattern histograms of sizes 256, 511, 512 and 256 bins, using the basic LBP, ILBP, LTP and MLBP, respectively. Given that the threshold t used to compute the LTP codes is defined as 5. For more details about the computation of these local pattern methods refer to Chapter 3 (sections 3.3 and 3.4).

— **Optimization:** It is an important phase in our proposed CBIR system. In this phase which runs offline, the weight vectors corresponding the database classes are generated. For this, the optimization database is first split into partial sets, in such a way that each partial set contains only images from the same class. Accordingly, the DE algorithm runs on each partial set instead of running on the global optimization database, such as in (Boukerma et al, 2019) . After initializing the control parameters of the DE algorithm (F and CR) and the population size, the algorithm iterates up to the predefined number of iterations (generations), so that at each iteration only the weight vector whose fitness is better than that of the previous iteration is maintained. In the case of our study, the initial population is randomly generated in a range of small values between 0 and 0,1. In fact, the population in this work represents the optimal weight vector to be associated to the local patterns; hence, its size is equal to the size of the local pattern histogram. In addition, the precision (APR) is considered as the fitness function in the DE process, and the objective is then to maximize it. The DE process runs iteratively on each partial optimization set until the maximum number of generations is reached. After

running DE on all partial sets, we obtain the optimal weight vectors corresponding to each class of the database.

— **Retrieval:** In this phase**,** which is an online phase, the optimal weight vectors generated during the optimization phase are used to weight the texture features extracted from images. After submitting a query image, it is first preprocessed by converting it into the gray level; then the texture features are extracted from it using the basic LBP method, then the ILBP, LTP and MLBP methods. The features extracted from the query image are subsequently compared to those extracted from the images of the test database using either Manhattan or Euclidean distance. However, to compute the distance between the features extracted from the query image and those of the test database images, each pattern constituting the feature is first weighted by its corresponding value in the optimal feature vector. The deal that arises here is how to find the appropriate optimal weight vector for the current query image? To address this, KNN algorithm is employed to determine the nearest class to the query image among the classes of images in the optimization database. The assumed class determined using KNN, allows us to select the corresponding optimal weight vector among all those generated during the optimization phase. Furthermore, the selected weight vector is refined and its values are readjusted according to the importance of the pattern in the image. The idea behind the readjustment technique used in this work is in fact inspired by the principle of the dominant local binary patterns (DLBP) (Liao et al, 2009), in which the occurrence frequencies of the patterns in the image are exploited to determine the dominating patterns in the image. The most frequently occurred patterns or dominant patterns are thus considered as the most important patterns in the image. Building on this idea, we readjust the values of the weight vector associated to the query image according to the occurrence frequency of the corresponding patterns. In this context, the bins of the local pattern histogram are first sorted in a descending order; then, the patterns whose occurrence number exceeds 0.75% of the total number of patterns in the image are considered as important patterns. The value of is this threshold (0.75%) is determined experimentally. Accordingly, the weight values corresponding to the important patterns are squared and the other values which correspond to the less important patterns remain unchanged. Mathematically, the readjustment of the weight values is defined in equations (5.6) and (5.7).

The Manhattan and Euclidean distances are then calculated as follows:

$$d_{MH}(u,v)=\sum_{i=1}^{D} p'_{c,i}|x_i - y_i| \tag{5.4}$$

$$d_{EUC}(u,v)=\sqrt{\sum_{i=1}^{D} (p'_{c,i}x_i - p'_{c,i}y_i)^2} \tag{5.5}$$

where u ($u=x_1, x_2, \ldots, x_D$) and v ($v=(y_1,y_2,\ldots,y_D)$ are two local pattern histograms computed for two different images; D is the dimension of the histograms;

Knowing that P_c ($P_c=(p_{c,1},p_{c,2},\ldots,p_{c,D})$ is the optimal weight vector corresponding to the assumed class c of the query image, the adjusted weight vector P'_c is defined as follows:

$$P'_c=\left(p'_{c,1},p'_{c,2},..,p'_{c,D}\right) \tag{5.6}$$

$$p'_{c,i}=\begin{cases} p_{c,i}^2 & if \quad occ\_rate > 0.75\% \\ p_{c,i} & otherwise \end{cases} \tag{5.7}$$

where occ_rate is the occurrence's rate of the pattern i in the image.

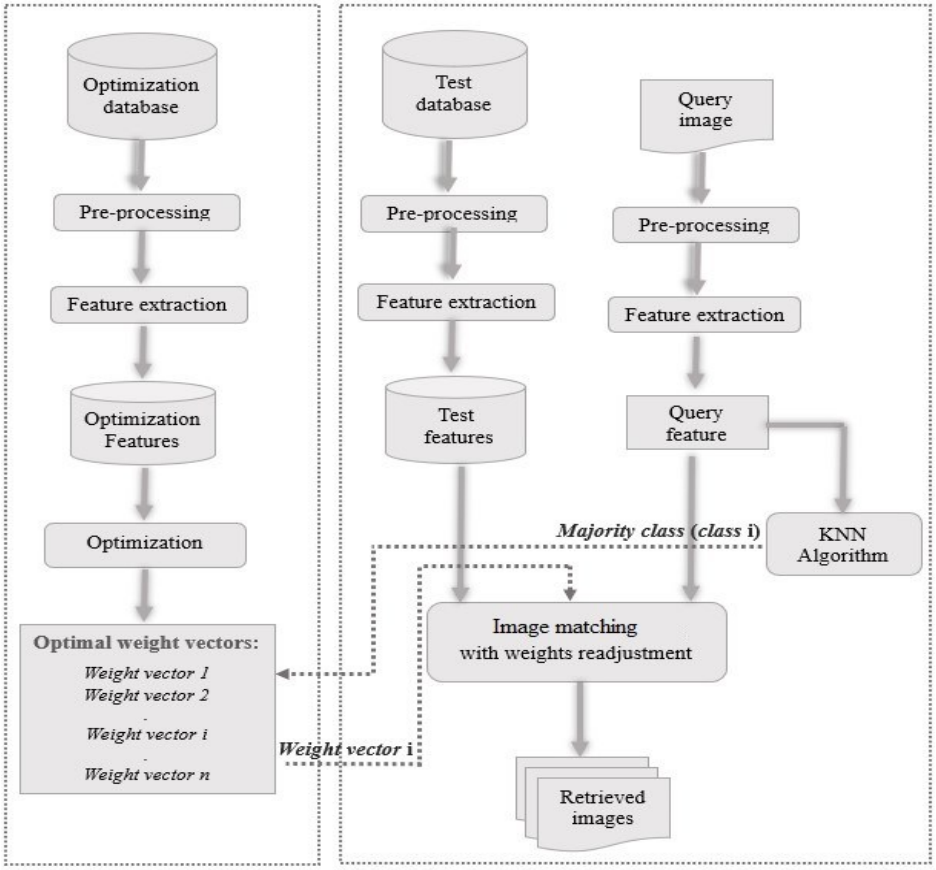


Figure 5. 6: Architecture of the proposed dynamic weighting approach

5.3.3 Experimental evaluation

In this section, we present the experimental framework used to evaluate the proposed weighting method.

5.3.3.1 Experimental setup

To implement the DPW approach we used the same system configuration and programming language employed in the previous contribution.

5.3.3.2 Parameters setting and dataset

In this work, we used the standard DE algorithm (the variant "DE / best /1 / bin) due to its success and superiority, compared to other variants such as Self-Adaptive Differential Evolution (SADE) (Qin et al, 2008), in both quality of the final result and rate of convergence (Georgioudakis & Plevris, 2020). To execute the chosen DE variant, we set the population size to 20 and the maximum number of generations to 50. Additionally, to achieve the best possible performance (APR), we used diverse values of the control parameters (F and CR) based on a well-defined test set (F=0.5, CR=0.9 / F=2, CR=1 / F=1.5, CR=0.9 / F=2, CR=0.1). After executing the DE algorithm using the different values of the test set, the weight vector which yields the best-known APR among all these executions is retained.

5.3.3.3 Results and analysis

Basically, the proposed dynamic weighted methods, namely Dynamic Weighted LBP (DWLBP), Dynamic Weighted ILBP (DWILBP), Dynamic Weighted LTP (DWLTP) and Dynamic Weighted MLBP (DWMLBP) are compared to the weighted methods proposed in the work of (Boukerma et al, 2019), which are WLBP, WILBP, WLTP and WMLBP, as well as the baseline methods (LBP, ILBP, LTP and MLBP). The comparison of the performance of these methods performance was carried using the average precision rate metric.

Table 5.8 presents a comparison by class between DWLBP, WLBP and basic LBP. The APR results presented in this table were obtained for the top 20 retrieved images, using the Manhattan and Euclidean distances.

Table 5.8: Comparison by class of the average precision rates between LBP, WLBP and DWLBP for the top 20 retrieved images (Boukerma et al, 2025a)

Distances	*Manhattan distance*			*Euclidean distance*		
Classes	*LBP (%)*	*WLBP (%)*	*DWLBP (%)*	*LBP (%)*	*WLBP (%)*	*DWLBP (%)*
Africa	54.69	**56.37**	55.37	49.37	55.75	**56.81**
Beach	47.62	53.62	**55.25**	35.81	49.00	**56.19**
Monument	41.19	43.06	**52.94**	29.44	40.81	**50.19**
Bus	91.19	**93.94**	93.06	75.87	93.87	**94.00**
Dinosaur	97.75	98.50	**99.19**	93.69	98.37	**99.25**
Elephant	35.44	41.94	**42.00**	28.62	37.12	**39.50**
Flower	81.06	**85.45**	84.31	64.69	81.94	**84.69**
Horse	68.25	64.94	**69.00**	63.62	64.19	**67.12**
Mountain	31.62	**33.00**	32.00	28.37	**33.12**	32.44
Food	49.69	50.69	**57.94**	40.37	51.44	**56.31**
Average	59.85	62.15	**64.11**	50.99	60.56	**63.65**

As observed in Table 5.8, the best APRs are obtained when using the DWLBP, either for Manhattan or Euclidean distance. For Manhattan distance, the DWLBP achieved an APR of 64,11% against 62,15% for WLBP and 59,85% for LBP. As for the Euclidean distance, the DWLBP reached an APR of 63,65% against 60,56% for WLBP (with 3.09% gain) and 50.99% against LBP. Furthermore, it is noteworthy that for almost all classes the DWLBP achieves the best APRs.

Further comparison of the APR between all proposed dynamic weighted methods, the weighted methods proposed in (Boukerma et al, 2019) and the baseline methods is provided in Table5.9.

Table 5.9: Comparative table of the average precision rates between the baseline, weighted and dynamic weighted methods for the 20 retrieved images (Boukerma et al, 2025a)

Method	*Manhattan*	*Euclidean*
LBP	59.85	50.99
WLBP	62.15	60.56
DWLBP	**64.11**	**63.65**
ILBP	63.12	53.22
WILBP	65.61	63.38
DWILBP	**65.76**	**65.27**
LTP	54.47	42.63
WLTP	61.76	60.39
DWLTP	**61.97**	**61.72**
MLBP	59.49	52.11
WMLBP	60.41	59.28
DWMLBP	**61.52**	**60.57**

The results presented in Table 5.9 clearly prove the superiority of the dynamic weighted methods over the weighted methods. In particular, for the Manhattan distance, the proposed DWLBP method could reach an APR of 64.11% against 59.85% for the WLBP. The proposed DWILBP in turns could reach an APR of 65.76% against 65.61% for the WILBP. In addition, the DWLTP achieved an APR of 61.97% against 61.76% for the WLTP. On the other hand, the DWMLBP reached an APR of 61.52% against 60.41% for the WMLBP. As for the Euclidean distance, the APR results of DWLBP, DWILBP, DWLTP and DWMLBP were considerably increased to 63.65%, 65.27%, 61.72% and 60.57%, with an increase of 3.09%, 1.89%, 1.33% and 1.29%, respectively.

5.4 Conclusion

This chapter outlined the first two contributions of our study, which lie in the development of discriminative and robust feature representations. By enhancing the robustness against illumination change of the existing OCLBP and IOCLBP methods, the proposed O-OCLBP and O-IOCLBP methods could achieve significant improvements in the retrieval performance, as demonstrated through the experimental evaluations. However, it should be noted that the proposed descriptors remain constrained by the high dimension of the features derived using these descriptors. Nonetheless, this limitation does not diminish the relevance and effectiveness of the proposed methods, but it addresses a critical need for future refinements. The second contribution, for its part, was able to enhance the representation of images by introducing an effective weighting mechanism that leverage an optimization technique, consisting in the DE algorithm, and a machine learning technique, consisting in the KNN algorithm. While the proposed weighting method demonstrates clear improvement compared to the existing methods, the retrieval performance of the LBP method and its gray level variants remain lower when used alone.

Chapter 6

Reducing Feature Dimensionality: Novel Feature Selection Methods for Effective and Efficient CBIR

6.1 Introduction

The results obtained in the previous experimental chapter confirm that the LBP method and its grayscale variants are highly discriminative and can effectively capture the texture information, but also that they remain less effective in terms of retrieval performance. On the other hand, incorporating color information with texture information can effectively improve the local pattern methods effectiveness. This was practically demonstrated in our first contribution. However, as it was highlighted, the color-texture LBP variants lead to a high-dimensional feature vector. Another way to improve the effectiveness of the local pattern methods is to extract the features using different local pattern methods then combine them in a single representation. However, this approach also leads to a high-dimensional feature space, which is computationally expensive. To address this issue, we introduce in this chapter three feature dimensionality reduction methods, designed specifically for histopathology image retrieval. Thus, in the third contribution (Boukerma et al, 2023) of this study, we propose to extract different ILBP features from multiple color spaces, then, we combined them to enhance the CBHIR performance. To reduce the dimension of the combined ILBP features, we introduce a feature selection scheme based on PSO algorithm to select the most discriminative patterns from the combined feature vector, thereby reducing its dimension. In the fourth contribution (Boukerma et al, 2025b) we expand the proposed method by using other local pattern methods, namely color LBP and color OCLBP, and by relying on SCA algorithm, which has better global search capability, to achieve a more adaptable dimensionality reduction method. Moreover, in addition to the KIMIA Path 960 histopathology image dataset used in the previous contribution, we use in this contribution a

larger and diverse histopathology image dataset, which is Kather-5k dataset. Finally, in the fifth contribution (Boukerma et al, 2024), we leverage deep learning capabilities, especially CNNs models, to extract effective deep features from colon histology images; then, we propose a dimensionality reduction method based on SCA algorithm to optimize the extracted and combined deep features.

6.2 A dimensionality reduction method based on particle swarm optimization algorithm

6.2.1 Overview

In medical image retrieval, and particularly content-based histopathology image retrieval (CBHIR), the extraction of suitable features plays a crucial role in the representation and interpretation of the histopathological images which represent different tissues. An effective and efficient CBHIR system can, in fact, facilitate the disease detection and diagnosis by clinicians. The specificity of histopathology images compared to other types of medical images is that histopathology images are typically color images, whereas other types of medical images are usually monochrome images, acquired from biopsies using computerized electron microscopes. The spatial organization of the cellular and connective tissue components in biopsy images as well as the repetitive patterns in this type of images make them highly appropriate for texture analysis (Cai et al, 2022). In this context, local pattern methods are widely used and successful in CBHIR (Khadilkar, 2022). On the other hand, as mentioned in Chapter 3 (section 3.2), local pattern methods are very appropriate for applications with limited data, and as histopathology image datasets are usually of small size, these methods can be effectively used for their analysis. Moreover, due to the fact that histopathology images are color images, the color information is also a significant characteristic in these images.

Given these considerations, incorporating color and texture to derive relevant features can effectively improve the performance of CBHIR. Therefore, we propose in this work to extract discriminative features from histopathology images using improved local binary pattern (ILBP) method. The ILBP features are computed in multiple color spaces, then combined in a single feature vector to enhance the retrieval performance. However, as the resulting feature vector is of high dimension, it can make the retrieving process very slow.

To address this issue, we propose a dimensionality reduction method based on PSO algorithm. The latter is applied in an offline phase to generate optimal weights to be used for the selection of most informative patterns among all the patterns constituting the combined local pattern histograms.

6.2.2 Proposed methodology

Knowing that D is the dimension of the combined ILBPs histogram and D′ is its dimension after reduction. The aim of our proposed method is to convert the D-dimensional feature into a lower D′-dimensional space. To achieve this objective, we use the PSO algorithm to generate a set of optimal weights that serve for selecting the most relevant patterns from the combined ILBPs histogram. Mainly, the proposed CBHIR system (depicted in Figure 6.1) consists of two major phases: feature extraction and dimensionality reduction.

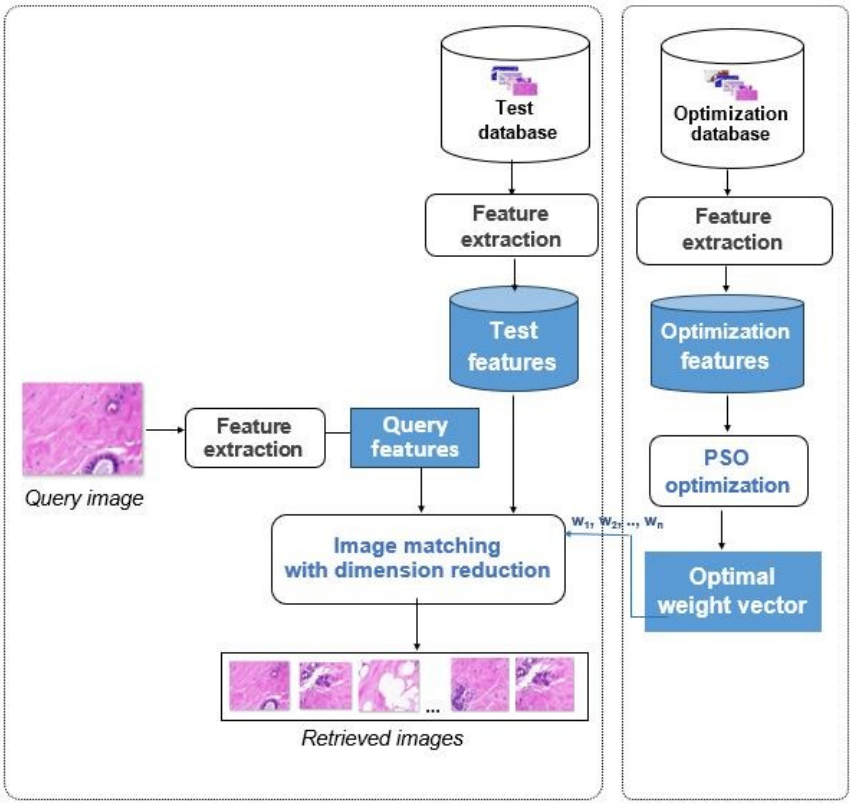


Figure 6. 1: Proposed CBHIR architecture (Boukerma et al, 2023)

Feature extraction: To enhance the CBHIR system performance, we extract effective and discriminative features from the colored histopathology images by capturing both color and texture information. In particular, we apply the grayscale $ILBP_{8,1}$ descriptor on each color channel of three color spaces, namely RGB, HSV and CIELAB. Subsequently, we combine

the resulting nine color ILBP histograms to form a single histogram of a dimension of 4599 bins (511×9). The combined color ILBP feature vector obtained from the three color spaces by applying the grayscale $ILBP_{8,1}$ (defined as in equations 3.10 and 3.11) in on the nine color channels (R, G, B, H, S, V, L, a and b), is given by the following equation:

$$\text{Combined} - \text{ILBP} = [\text{ILBP}_{\text{R}}, \text{ILBP}_{\text{G}}, \text{ILBP}_{\text{B}}, \text{ILBP}_{\text{H}}, \text{ILBP}_{\text{S}}, \text{ILBP}_{\text{V}}, \text{ILBP}_{\text{L}}, \text{ILBP}_{\text{a}}, \text{ILBP}_{\text{b}}] \quad (6.1)$$

Dimensionality Reduction: In this phase, the resulting combined color ILBP features extracted from all images in Kimia Path960 database (whether those of the optimization or testing set) are reduced. In a first step, the optimal weight vector is generated, then, it is used to reduce the feature dimension.

To generate the optimal weight vector, the PSO algorithm is applied on the optimization set of the image database, which contains the first 20% images from the total number of images in the database. In this first step, which runs offline, the PSO algorithm starts by initializing the population of particles which is generated randomly in the range [0,1]. The dimension the population is therefore equal to that of the combined color ILBP histograms (4599 bins). Thereafter, the fitness of each particle is calculated. Here, we consider the average precision rate (APR) as well as the average recall rate (ARR) as fitness function of our optimization problem. The aim is then to maximize either the APR or ARR. After computing the fitness of all particles, the best positions ever found for each particle at the current iteration is updated. This process is repeated until the maximum number of iterations is reached. Among the pbest values of all particles, the particle with the best value, which corresponds, indeed, to the highest APR or ARR, is retained and stored as the global best position—gbest—. This process is iterated until a predetermined number of iterations is reached.

After the optimization step which results in the generation of the optimal weight vector, the next step is the reduction of the combined ILBP feature dimension. This step is performed online during the retrieval process. Specifically, when a user submits a query image, the nine color ILBP features are extracted from the query image, then combined in a single feature vector. The latter is then compared to the combined color ILBP features of all images in the test set of the KIMIA Path 960 database, based on a chosen matching measure. More precisely, the comparison is performed between the reduced feature vectors (containing only the selected patterns) of the query image and all the test set images. To obtain the reduced

features, we use the weight vector generated during the optimization step in such a way that only the patterns whose corresponding weight values are higher than 0.5 are selected, while other patterns less important are discarded. The threshold 0.5 is a predetermined value determined experimentally. Thereafter, the distances between the reduced features of the query image and those of the test set images are calculated while multiplying each value of the selected patterns by its corresponding value of the weight vector. Finally, the distances are sorted in descending order, and the top 20 images are displayed to the user as a response to its query image.

6.2.3 Experimental evaluation

6.2.3.1 Experimental details

In this section, we provide the details of the carried-out experiments, including experimental setup and other important parameters.

— **Experimental setup and parameters setting:** In this work, we used the standard version of the PSO algorithm with the following parameters: w=1, c1=3 and c2=1. In addition, we set the population size to 20 and the maximum number of generations to 50. To implement the proposed method, we used MATLAB language on an Intel® Core i7 with 2.6 GHz CPU and 8 GB internal RAM.

— **Other parameters:** As we have previously mentioned, we used the average precision rate (APR) and the average recall rate (ARR) as performance metrics, the Manhattan distance as a matching measure. As for the image database, we have used KIMIA Path960 which consists of 960 histopathology images from 20 different tissue types. For more details about this dataset, refer to Chapter 2 (section 2.7.2).

6.2.3.2 Results and analysis

A. Comparison of the performance and feature dimension

In this section, we present the carried-out experiments while analyzing the obtained results. The best results obtained from the different experiments are highlighted in bold.

Table 6.1 provides a comparison of the performance, in terms of APR and ARR, and the feature size between the proposed method (reduced feature vector), the combined color ILBP features (Combined-color ILBP) and the color ILBPs computed in each single-color

space ($ILBP_{RGB}$, $ILBP_{HSV}$ and $ILBP_{CEILAB}$). The APR and ARR results were obtained for the top 20 retrieved images, using Manhattan distance.

Table 6.1: Performance and feature size comparison for the top 20 images using Manhattan distance (Boukerma et al, 2023)

Method	Color space	Precision		Recall	
		Feature size	*APR (%)*	*Feature size*	*ARR (%)*
$ILBP_{RGB}$	RGB	1533	77.74	1533	40.92
$ILBP_{HSV}$	HSV	1533	74.44	1533	39.18
$ILBP_{CEILAB}$	CEILAB	1533	80.13	1533	42.17
Combined-color ILBP	RGB-HSV-CEILAB	4599	85.76	4599	45.13
Proposed method	RGB-HSV-CEILAB	2306	**86.82**	2229	**45.60**

As shown in Table 6.1, the best APR and ARR results were obtained when using the proposed method, with an APR of 86.82% against 85.76% for the combined color ILBP (without reduction); while the APRs achieved by the ILBP methods computed in a single-color space were only 77.74%, 74.44% and 80.13% for $ILBP_{RGB}$, $ILBP_{HSV}$ and $ILBP_{CEILAB}$, respectively. Similarly, the ARR results of the proposed method are better than the other methods, with 45.60% against 45.13% for the combined color ILBP, and 40.92%, 39.18%, 42.17% for the $ILBP_{RGB}$, $ILBP_{HSV}$ and $ILBP_{CEILAB}$, respectively. Furthermore, it is evident from the table that the feature dimension of the proposed method (features with reduction) is notably lower than the combined color ILBP (without reduction), with a reduction gain of about 50%, either when using the APR as a fitness function and ARR).

B. Comparison of the retrieval time

To evaluate the efficiency of the proposed method, we compare the average retrieval time taken when using the reduced feature with that taken when using the full-dimensional feature (combined color ILBP). These average retrieval time values were calculated for the top 20 retrieval.

The chart on Figure 6.2 depicts the average retrieval time of the combined color ILBP method and the proposed method.

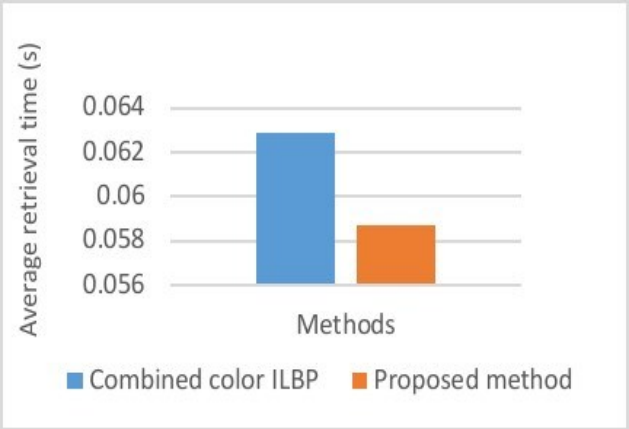


Figure 6. 2: Comparison of the average retrieval time between the reduced and full-dimensional features (Boukerma et al, 2023)

According to the data in the chart, the average retrieval time of the proposed method, in which the reduced features are used for the retrieval, is shorter than that obtained when using the combined color ILBP features, with 0.0587 sec against 0.0629 sec.

6.3 An enhanced pattern selection scheme based on sine cosine algorithm as a tool for dimensionality reduction

6.3.1 Overview

To take greater advantage of the capability of local pattern methods in representing the tissue patterns of histopathology scans, we propose in this work to use more discriminative and effective local pattern methods to extract relevant features from more than one histopathology image dataset. In particular, we use combined color LBP and Combined color OCLBP features in representing the histopathology images of KIMIA Path960 and Kather-5K histopathology datasets.

To reduce the size of the combined features, we propose an offline optimization mechanism based on SCA algorithm, since it is less sensitive to parameter tuning than PSO, with a strong exploration capability and fast convergence. In this work, as the previous contribution, we employ feature selection based on optimization algorithms for reducing the dimension of the extracted features. This direction was driven by the fact that, when relying on optimization algorithms, the only requirement is to find the optimal parameters of the model which maximize or minimize the fitness, and thereby guide the search process to the best solution (Thirumalaisamy et al, 2023). Achieving the optimal parameters using the

training data enables the model to be applied effectively on the test data (the novel or unknown data). On the other hand, although deep learning models exhibit high performance on the training data, they may underperform on the test data where the model could fail to generalize (Thirumalaisamy et al, 2023). This concern, which is in fact related to the overfitting issue, becomes significantly more complex in the case of medical applications where the image datasets are usually of a limited size. In fact, this situation arises from the fact that, in general practice, it is challenging to collect considerable amounts of medical images, specifically for low-incidence serious diseases (Kim et al, 2019).

Based on these considerations, we opt to explore the use of optimization algorithms for feature selection, as a tool for feature dimensionality reduction. In particular, we adopt a pattern selection approach to select the most relevant patterns among all those constituting the combined local pattern histograms.

6.3.2 Proposed methodology

In this section we explain in detail our proposed approach which consists in a multi-color space pattern selection (MCSPS) method. The proposed MCSPS approach starts by extracting the local pattern features from the images of each dataset. To compute the color LBP histograms, the basic LBP is applied on the nine color channels of the multi-color space, which consists in three color spaces, namely RGB, HSV and YCbCr. The obtained color LBP histograms are then concatenated to form the Combined-LBPs vector with 2304 bins. Likewise, the Combined-OCLBPs vector is obtained by concatenating eighteen color OCLBP histograms obtained by applying the basic LBP independently and jointly on the nine color channels of the three-color space. The Combined-OCLBPs vector consists then of 4608 bins. To reduce the dimension of the two combined local pattern features, we apply our offline optimization scheme based on SCA algorithm to generate the optimal weights that allow selecting the most relevant patterns from these features Furthermore, these optimal weights are subsequently used to weight the selected patterns.

Specifically, our proposed CBHIR system consists mainly of three phases: feature extraction, optimization and retrieval.

Feature extraction: As mentioned above capturing both color and texture information may improve considerably the retrieval performance. Moreover, capturing the color information from a hybrid color space is more effective than capturing it from one particular color space

(Porebski et al, 2018). For this reason, we propose to extract the color local patterns from a hybrid color space composed of three color spaces, which are RGB, HSV and YCbCr. Therefore, to extract the color LBP features, we apply the $LBP_{8,1}$ operator on each color channel of the three-color spaces, namely the channels: R, G, B, H, S, V, Y, Cb and Cr. The application of LBP on a specific color channel is defined in equation (6.1). Thereafter, the obtained nine color LBP histograms are concatenated to form the combined-LBPs feature, as defined in equation (6.2). Similarly, to obtain the color OCLBP features, the $LBP_{8,1}$ is first applied on each color channel independently (RR, GG, BB, HH, SS, VV, YY, CbCb, CrCr) to derive the OCLBP intra-channel feature. Then, it is applied on pairs of color channels jointly (RG, RB, GB, HS, HV, SV, YCb, YCr and CbCr) to derive the OCLBP inter-channel features. The general form of the computation of OCLBP on a specific pair of color channels is expressed as in equation (6.3). After the derivation of the eighteen color OCLBP histograms, they are subsequently concatenated to form the combined-OCLBPs, which is defined as in equation (6.4).

Given a neighborhood of P pixels at a radius R, the LBP code of a pixel c for a color channel M is defined as follows:

$$LBP_{(P,R)_M}(x_{c,M},y_{c,M})=\sum_{p=0}^{P-1} S(g_{p,M}-g_{c,M})2^p \tag{6.1}$$

$$\begin{aligned} &Combined-LBPs \\ &=[LBP_R, LBP_G, LBP_B, LBP_H, LBP_S, LBP_V, LBB_Y, LBP_{Cb}, LBP_{Cr}] \end{aligned} \tag{6.2}$$

Given a neighborhood of P pixels at a radius R, the OCLBP code of a pixel c for a pair of color channel (M,N) is defined as follows:

$$OCLBP_{(P,R)_{M,N}}(x_{c,M},y_{c,M})=\sum_{p=0}^{P-1} S(g_{p,N}-g_{c,M})2^p \tag{6.3}$$

$$\begin{aligned} &Combined-OCLBPs \\ &=\begin{bmatrix} OCLBP_{RR}, OCLBP_{GG}, OCLBP_{BB}, OCLBP_{RG}, OCLBP_{RB}, OCLBP_{GB}, \\ OCLBP_{HH}, OCLBP_{SS}, OCLBP_{VV}, OCLBP_{HS}, OCLBP_{HV}, OCLBP_{SV}, \\ OCLBP_{YY}, OCLBP_{CbCb}, OCLBP_{CrCr}, OCLBP_{YCb}, OCLBP_{YCr}, OCLBP_{CbCr} \end{bmatrix} \end{aligned} \tag{6.4}$$

Optimization: To reduce the dimension of the combined histograms derived in the previous phase, we apply the SCA algorithm on the optimization set of each image database, which contains the first 20% images of the overall database. Here, the role of the SCA is to learn

the optimal weights that serve to select the most discriminative patterns from the derived local pattern features. Accordingly, in this phase, which runs offline, the SCA starts by generating randomly the initial population of D individuals in the range [0, 0.1]. Where, D is the dimension of the combined local patterns' histogram (2304 bins for the combined-LBPs and 4608 bins for the combined-OCLBPs). Each potential solution or population individual is in fact the weight vector that maximizes the retrieval performance. The average precision rate (APR) is considered as the fitness function. In addition, to evaluate and confirm the efficiency of our proposed scheme, we also used in another experiment the average recall rate (ARR) as a fitness function. The generated solutions are then evaluated to determine the best solution. Then, the individuals' position and the random parameters are updated. This process iterates until the maximum number of iterations is reached. The final step of the algorithm is to memorize the best global solution found among all known solutions, as well as the best fitness value. The pseudo-code of the SCA process is provided in the following algorithm.

Algorithm 1 Pseudo-code of sine cosine algorithm

1 Define the population size (N) and the maximum number of iterations (max_iter)
2 Generate randomly an initial population X of size N × D, Within the range [0, 0.1]
3 Initialize the random parameters r_1, r_2, r_3 and r_4
4 Calculate the fitness values (APR or ARR) for each individual
5 Choose the best solution which has the maximum fitness values and save its position
6 **for** iter = 1:max_iter d**o**
7 Update r_1, r_2, r_3 and r_4
8 **if** $r_4 < 0.5$
9 Update the individuals' position by applying equation (4.9)
10 **else**
11 Update the individuals' position by applying equation (4.9)
12 **end if**
13 Calculate the fitness values for each updated position
14 Update the best solution and its position
15 **end for**
16 Return the best solution which provides the global best fitness values

Retrieval: Upon completion of the optimization phase, the generated optimal weight vector is used to select the most relevant patterns from the patterns extracted from images. In this phase, which performs online, the size of the combined local pattern features extracted from the images are reduced by selecting only the most relevant patterns. In particular, the optimal weights are associated with the corresponding patterns of the combined histograms when calculating the distances between query image and all images in the test set of the image dataset, which consists of 80% of the overall images of the dataset. The size of each histogram is then reduced by maintaining only the bins that correspond to the non-zero values of the optimal weight vector, the other bins are discarded. The non-zero values of the weight vector are then multiplied by the corresponding bins of the histogram when calculating the distances. Accordingly, the dimension of the combined color local pattern histograms is reduced while at the same time weighted.

6.3.3 Experimental evaluation

After describing the proposed methodology, this section presents the conducted experiments followed by a comprehensive analyze and discussion of the obtained results.

6.3.3.1 Experimental details

This section provides a detailed description of the experimental environment used to implement and evaluate our proposed method.

— **Experimental setup:** Extensive experiments were conducted on a machine equipped with an Intel i7-5600U CPU at 2.60 GHz and 8 GB internal RAM. The proposed method was implemented using MATLAB language.

— **Datasets:** The performance of the proposed method was evaluated on two histopathology image databases, namely KIMIA Path960 and Kather-5K. The former consists of 960 images of different tissue types grouped into 20 classes, and the latter contains 5000 images representing different tissues of human colorectal cancer grouped into 8 classes. Further details and samples of each dataset are provided in Chapter 2, section 2.7.2.

— **Parameters setting:** In the optimization phase, we particularly used the standard SCA for generating the optimal weight vector, we set the population size to 20 individuals and the number of generations to 50 generations. In addition, we used various distance metrics to compute the distance between images, including Manhattan, Euclidean, D1 and Square chord

distances. To assess the performance of the proposed method, we used the APR and ARR, which are widely used metrics in CBIR for the performance evaluation.

6.3.3.2 Results and analysis

To thoroughly assess the effectiveness and efficiency of the proposed MCSPS method, a series of experiments were conducted on the two histopathology image datasets.

A. Evaluation of the method effectiveness

To evaluate the effectiveness of the proposed method, the APR and ARR are computed on both datasets.

Tables 6.2 and 6.3 display the APR and ARR results obtained on KIMIA Path 960 and Kather-5K databases, respectively, using various distance metrics, including Manhattan, Euclidean, D1 and Square chord. In particular, the proposed multi-color space pattern selection method for LBP (MCSPS-LBP) is compared to the LBP methods derived from a single-color space (LBP_{RGB}, LBP_{HSV} and LBP_{YCbCr}), and the proposed multi-color space pattern selection for OCLBP (MCSPS-OCLBP) is compared to the OCLBP methods derived also from a single-color space ($OCLBP_{RGB}$, $OCLBP_{HSV}$ and $OCLBP_{YCbCr}$). The combined-LBPs and Combined-OCLBPs were, however, derived from the three-color spaces RGB, HSV and YCbCr.

Table 6.**2**: Retrieval performance on Kimia Path960 database for the top 20 images (Boukerma et al, 2025b)

Method	*Manhattan*		*Euclidean*		*D1*		*Square chord*	
	APR (%)	*ARR (%)*	*APR (%)*	*ARR (%)*	*APR (%)*	*ARR (%)*	*APR (%)*	*ARR (%)*
LBP_{RGB}	73.95	38.92	68.25	35.92	77.78	40.93	74.48	39.20
LBP_{HSV}	78.93	41.54	74.93	39.43	81.04	42.65	78.98	39.20
LBP_{YCbCr}	78.43	41.28	70.56	37.14	81.72	43.01	78.82	41.48
Combined-LBPs (full feature set)	80.66	42.45	76.69	40.36	83.60	44.00	81.21	42.74
Proposed MCSPS-LBP (reduced feature set)	**83.26**	**43.48**	**81.52**	**42.87**	**84.03**	**44.22**	**83.38**	**43.40**
$OCLBP_{RGB}$	84.58	44.51	74.87	39.41	85.58	45.04	84.49	44.47
$OCLBP_{HSV}$	83.97	44.19	71.54	37.65	86.37	45.46	85.85	45.19
$OCLBP_{YCbCr}$	84.43	44.43	73.87	38.88	87.31	45.95	85.34	44.92
Combined-OCLBPs (full feature set)	88.28	46.46	81.66	42.98	91.14	47.97	88.66	46.66
Proposed MCSPS-OCLBP (reduced feature set)	**89.70**	**47.07**	**87.68**	**45.22**	**91.18**	**48.11**	**90.17**	**47.38**

Table 6.3: Retrieval performance on Kather-5K database for the top 20 images (Boukerma et al, 2025b)

Method	*Manhattan*		*Euclidean*		*D1*		*Square chord*	
	APR (%)	*ARR (%)*	*APR (%)*	*ARR (%)*	*APR (%)*	*ARR (%)*	*APR (%)*	*ARR (%)*
LBP_{RGB}	65.81	2.63	63.03	2.52	63.65	2.54	66.35	2.65
LBP_{HSV}	72.45	2.90	70.43	2.82	70.04	2.80	73.54	2.94
LBP_{YCbCr}	71.72	2.87	69.30	2.77	65.01	2.60	73.57	2.94
Combined-LBPs (full feature set)	74.06	2.96	73.51	2.94	71.26	2.85	74.87	2.99
Proposed MCSPS-LBP (reduced feature set)	**75.61**	**2.99**	**73.80**	**2.95**	**73.31**	**2.94**	**77.35**	**3.08**
$OCLBP_{RGB}$	78.41	3.14	72.53	2.90	77.66	3.11	79.15	3.17
$OCLBP_{HSV}$	80.71	3.23	75.38	3.01	81.61	3.26	82.44	3.30
$OCLBP_{YCbCr}$	79.60	3.18	74.20	2.97	79.58	3.18	82.10	3.28
Combined-OCLBPs (full feature set)	85.31	3.41	83.07	3.32	86.32	3.45	86.25	3.45
Proposed MCSPS-OCLBP (reduced feature set)	**85.44**	**3.42**	**83.22**	**3.35**	**86.37**	**3.45**	**86.85**	**3.48**

As observed from the two tables, the proposed MCSPS-OCLBP demonstrates superior performance over all methods, either in terms of APR or ARR, and across both KIMIA PATH 960 and Kather-5K databases and for all distances. Compared to the combined-LBPs, it is observed from Table 6.2 that the APR results of the proposed MCSPS-LBP on KIMIA PATH 960 dataset using Manhattan, Euclidean, D1 and Square chord distances were increased to achieve 83.26 %, 81.52%, 84.03% and 83.38%, with a gain of 2.60 %, 4.83%, 0.43% and 2.17%, respectively. The MCSPS-OCLBP method achieves also the best APRs using Manhattan, Euclidean, D1 and square chord with 89.70%, 87.68%, 91.18% and 90.17%, and a gain of 1.42%, 6.02%, 0.04% and 1.51%, respectively, compared to the combined-OCLBPs. As for Kather-5K dataset, the APR results of the proposed MCSPS-LBP presented on Table 6.3 indicate its superiority over the combined-LBPs, with APRs of 75.61%, 73.80%, 73.31% and 77.35%, and a gain of 1.55%, 0.29%, 2.05% and 2.48% using Manhattan, Euclidean, D1 and Square chord, respectively. The APR of the proposed MCSPS-OCLBP method for the same database was also superior compared the Combined-OCLBPs, with 85.44%, 83.22%, 86.37% and 86.85% versus 85.31%, 83.07%, 86.32% and 86.25%, when using Manhattan, Euclidean, D1 and Square chord distance, respectively.

B. Evaluation of the method efficiency

Tables 6.4 and 6.5 provide a comparison of the feature dimension between the proposed MCSPS-LBP and MCSPS-OCLBP methods on one hand and the combined-LBPs and combined OCLBPs on the other hand. The comparison was performed on KIMIA Path 960 database (Table 6.4) and Kather-5K database (Table 6.5). It is worth noting that these results were computed by considering the APR as the fitness function of the optimization problem.

Table 6.4: Number of features on Kimia Path960 database for the top 20 images when considering the APR as fitness function (Boukerma et al, 2025b)

Method	*Manhattan*	*Euclidean*	*D1*	*Square chord*
Combined-LBPs (full feature set)	2304	2304	2304	2304
Proposed MCSPS-LBP (reduced feature set)	**955**	**821**	**1018**	**1096**
Combined-OCLBPs (full feature set)	4608	4608	4608	4608
Proposed MCSPS-OCLBP (reduced feature set)	**1938**	**2048**	**1678**	**1918**

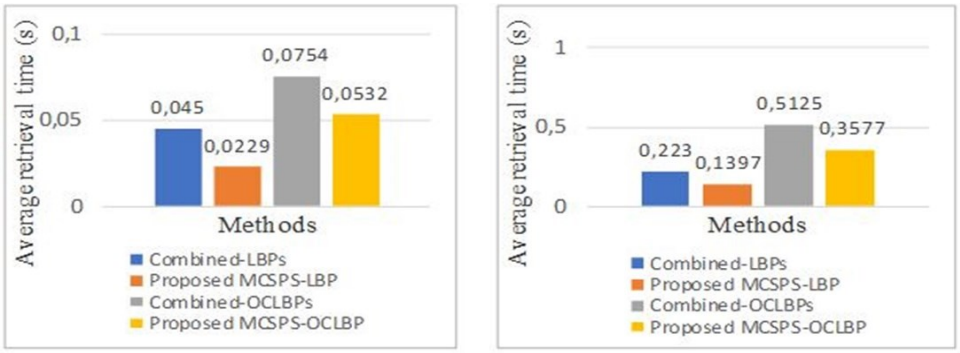
Table 6.5: Number of features on Kather-5K database for the top 20 images when considering the APR as fitness function (Boukerma et al, 2025b)

Method	*Manhattan*	*Euclidean*	*D1*	*Square chord*
Combined-LBPs (full feature set)	2304	2304	2304	2304
Proposed MCSPS-LBP (reduced feature set)	**1061**	**1028**	**906**	**1100**
Combined-OCLBPs (full feature set)	4608	4608	4608	4608
Proposed MCSPS-OCLBP (reduced feature set)	**1976**	**1689**	**1934**	**1783**

Tables 6.4 and 6.5 provide clear evidence that the improvement of the APR is accompanied by a reduction in the feature dimensionality. The dimension of the proposed methods was considerably reduced compared to the combined methods (full feature sets) across both image databases and for all used distances. From Table 6.4, we can observe that the proposed MCSPS-LBP dimension was reduced from 2,304 bins to 955, 821, 1,018 and 1,096 bins when using Manhattan, Euclidean, D1 and Square chord Distances, respectively; with a reduction percentage of 59%, 64%, 56% and 53 %, respectively. Likewise, instead of the 4,608 bins obtained when using the combined-OCLBPs, the proposed MCSPS-OCLBP could achieve only 1,938, 2,048, 1,678 and 1,918 bins when using the Manhattan, Euclidean, D1 and square chord distances, respectively. On the other hand, as shown in Table 6.5, the proposed MCSPS-LBP and MCSPS-OCLBP still maintains a reduced number of features on Kather-

5K database, compared to the combined-LBPs and combined-OCLBPs methods, respectively. In particular, the dimension of the proposed MCSPS-LBP was decreased from 2,304 bins to 1,061, 1,028, 906 and 1,100 bins, when using Manhattan, Euclidean, D1 and square chord distances, respectively; with a reduction percentage of 54%, 55%, 61% and 52%, respectively. Likewise, instead of 4,608 bins obtained when using the combined-OCLBPs, the proposed MCSPS-OCLBP could achieve 1,976, 1,689, 1,934 and 1,783 bins, with a reduction percentage of 57%, 63%, 58% and 61%, when using the Manhattan, Euclidean, D1 and square chord distances, respectively.

To assess the influence of the feature dimension reduction on the retrieval time, a comparison of the average retrieval time was conducted between the two proposed MCSPS methods and the combined methods. The obtained results for the retrieval of the top 20 images across both KIMIA PATH 960 and Kather-5K datasets and using Square chord distance are depicted in Figure 6.3.



(a) (b)

Figure 6. 3: Comparative chart of the average retrieval time: (a) KIMIA Path 960, (b) Kather-5K (Boukerma et al, 2025b)

The data presented on Figure 6.3 confirms the effectiveness of the proposed methods with regard to average retrieval time. The results clearly indicate that the average retrieval time taken by the proposed MCSPS-LBP is considerably shorter than that taken by the combined-LBPs method (0.0229 sec versus 0.0450 sec for KIMIA Path 960 database, and 0.1397 sec versus 0.2230 sec for Kather-5K database). Likewise, the average retrieval time taken by the proposed MCSPS-OCLBP was decreased from 0.0754 sec to 0.0532 sec for KIMIA Path 960 dataset, and from 0.5125 sec to 0.3577 sec for Kather-5K dataset.

C. Comparison with other state-of-the-art methods

— **Comparison with deep learning methods**

To further evaluate the effectiveness of our proposed optimization mechanism, the latter was applied on the features extracted using some CNN methods. In particular, the deep features were extracted using three pre-trained CNN models, namely AlexNet (Krizhevsky et al, 2012), MobileNetV2 (Sandler et al, 2018) and VGG16 (Simonyan & Zisserman, 2014). The deep features were in fact extracted from both KIMIA Path 960 and Kather-5K datasets using the same partitioning ratio adopted for the proposed methods (i.e., 20% for training and 80% for testing). In addition, to extract the deep features using these pre-trained CNNs, we applied transfer learning was applied on them, so that the features were extracted from the last CNNs layers: fc7 (for AlexNet), global_average_pooling2d_1 (for Mobilenetv2) and fc7 (for VGG16). The number of deep features extracted from these layers was in fact: 4,096, 1,280 and 4,096 for AlexNet, MobileNetV2 and VGG16, respectively. Table 6.6 outlines a comparison of both retrieval performance (in terms of APR) and feature dimension between the proposed methods and the CNN-based methods. For this experiment, the retrieval results were computed for the top 20 images using Manhattan and D1 distances.

Table 6.6: Comparison of the feature dimension and the retrieval performance between the proposed methods and some deep learning-based methods (Boukerma et al, 2025b)

Dataset	Method	Manhattan		D1	
		No. of features	APR	No. of features	APR
Kimia Path960	AlexNet	4096	85.63	4096	70.43
	MobileNetV2	1280	80.89	1280	85.03
	VGG16	4096	83.06	4096	77.12
	Proposed MCSPS-LBP	**955**	83.26	**1018**	84.03
	Proposed MCSPS-OCLBP	1938	**89.70**	1678	**91.18**
Kather-5K	AlexNet	4096	88.02	4096	83.32
	MobileNetV2	1280	83.35	1280	84.79
	VGG16	4096	**89.55**	4096	85.56
	Proposed MCSPS-LBP	**1061**	75.61	**906**	73.31
	Proposed MCSPS-OCLBP	1976	85.44	1934	**86.37**

It is obvious from Table 6.6 that the proposed MCSPS-LBP method exhibits lower feature dimension compared to the deep learning-based methods, for both databases and using both distances. As for the performance, it is observed that the APR of the proposed MCSPS-

OCLBP method achieve the best APR over almost all CNN models, except for Kather-5K database where the APR of VGG16 is better with Manhattan distance. On the whole, our proposed methods, and specifically the MCSPS-OCLBP method, have proved a significant efficiency over most tested CNN-based methods, whether in terms of feature dimensionality or performance.

— **Comparison with other handcrafted methods**

Table 6.7 presents a comparison of the APR results between the proposed MCSPS-OCLBP method and some handcrafted methods from the literature, using Manhattan distance. It should be noted that these state-of-the-art methods were applied on a different partitioning ratio compared to that used in this work. In fact, in (Yazdi & Erfankhah, 2020) the partitioning was random with 70% and 30% for the training and test sets, respectively, on Kather-5K dataset. In (Sukhia et al, 2019),(Dubey et al, 2016),(Lan & Zhou, 2016), 50% of Kimia Path960 images were randomly affected to the training set, 25% to the validation set and the remaining 25% were assigned to the test set. Accordingly, and since works on CBHIR remain limited, especially works with the same partitioning used in our work (20%for optimization and 80% for testing), we then re-applied our scheme using a partitioning score of 70%-30% (i.e., 70% for optimization and 30% for testing) for Kather-5K to be compared to the work of (Yazdi & Erfankhah, 2020). Additionally, and since we did not use a validation set in our work, we adopted a partitioning of 75% – 25% for Kimia Path960 to be compared to the works of (Sukhia et al, 2019),(Dubey et al, 2016),(Lan & Zhou, 2016). Nevertheless, we also compare the proposed MCSPS-OCLBP method obtained using our initial partitioning (20%–80%) with the above-mentioned state-of-the-art methods. The comparison is provided in Table 6.7. It should be noted that the results of MDLBP and CTS methods are reported from the paper of (Sukhia et al, 2019).

Table 6.7: Comparison of the average precision rates (%) between the proposed method and some state-of-the-art methods (Boukerma et al, 2025b)

Dataset	Method	Top 1	Top 5	Top 10	Top 15	Top 20
Kimia Path960	MDLBP (Dubey et al, 2016)	/	81.62	75.33	68.82	62.73
	CST (Lan & Zhou, 2016)	/	44.88	40.44	35.12	28.81
	LTP+VLAD (Sukhia et al, 2019)	/	93.56	88.54	83.02	78.95
	Proposed MCSPS-OCLBP (75% -25%)	100	92.58	84.21	65.00	51.27
	Proposed MCSPS-OCLBP (20% -80%)	**100**	**95.21**	**92.60**	**91.18**	**89.70**

Kather-5K	Riesz+M-LBP (Yazdi & Erfankhah, 2020)	90.82	86.73	83.60	/	79.72
	Proposed MCSPS-OCLBP (70% -30%)	100	90.16	85.60	81.82	79.62
	Proposed MCSPS-OCLBP (20% -80%)	**100**	**93.00**	**89.6**	**86.63**	**85.44**

As shown in the first part of the table, with regard to Kather-5K dataset, the proposed MCSPS-OCLBP with the partitioning ratio of 70%–30% exhibit mostly superior performance than the other state-of-the-art methods, with the exception of the method Riesz + M-LBP method, when applied for the top 20 retrieved images, where its APR exceeded the proposed method by a slight margin. With respect to Kimia Path960 dataset, it can be seen that the APR of the proposed method, with the partitioning score of 75% – 25%, surpassed that of the MDLBP and CST methods in most cases; however, it exhibits lower APR than LTP+VLAD. This can be explained by the fact that extracting features by relying on just one local pattern descriptor (such as LBP, OCLBP and MDLBP) does not generally result in high performance on a small image dataset. A case in point is when the Kimia Path960 was split with a ratio of (75%–25%), where the testing data consists only of 240 images. By contrast, when the image dataset was split with ratio of (20%–80%), the proposed MCSPS-OCLBP method yields superior APR over all the state-of-the-art methods, for both image databases. In summary, the analysis of the findings presented in this section indicates that our proposed methods exhibit higher performance when the optimization (training) set is considerably smaller than the test setting, which may be viewed as a strength as it enables handling the maximum amount of unknown data.

6.4 Optimized feature space based on deep learning and sine cosine algorithm

6.4.1 Overview

Building upon the findings and discussions in the preceding chapter, local pattern methods are highly appropriate for the retrieval of histopathology images. Nonetheless, the effectiveness of these methods decreases as the number of samples in the dataset increases. The results presented in the preceding chapter indicate clearly that the APR and ARR results achieved by all the methods on Kimia Path 960 are significantly better than those obtained when using Kather-5k dataset. Indeed, when the image dataset is relatively large, deep learning-based methods may be the most effective solution for the extraction of relevant

features from histopathology images. In particular, CNN-based methods have recently demonstrated encouraging performance results in the field of CBHIR (Alizadeh et al, 2023).

CNN models are generally pre-trained on very large image databases for different tasks. To adapt these pre-trained models to small image databases, generally collected for specific domains like biomedicine, transfer learning technique is commonly used. This technique is widely used today in CBIR to extract discriminative features. However, despite the discriminative power and effectiveness of these features, they are usually high-dimensional and lead to a high computational cost. To take advantage of the capabilities of deep learning-based methods in extracting effective features while ensuring lower computational cost, we propose in this work to extract deep features from histopathology images of Kather-5k dataset by applying transfer learning. The latter, is particularly applied on the pre-trained ResNet18 and GoogLeNet CNN models. The extracted deep features through these two models are subsequently combined to improve the retrieval performance. To reduce the dimension of the combined deep features, we propose to use an optimization mechanism based on SCA algorithm.

6.4.2 Proposed methodology

Despite the significant role of the manual analysis of histology images for pathologists, the automated processing of this type of images can provide a quantitative analysis with notable effectiveness, that can assist the pathologists in the diagnosis of various diseases such as colorectal cancer (Kather et al, 2016). CBIR is one of the computer-aided diagnosis tools that can assist pathologists in the automatic diagnosis and analysis of histology images. In particular, CBHIR systems are designed to locate and search images from a given histology image database, according to their similarity to the query image. In this work, we propose a CBHIR system for colon cancer histology images based on the extraction, combination and optimization of high-level features. The proposed system is primarily based on three essential phases, including feature learning, optimization and retrieval (see Figure 6.4). These phases are described in detail in the following sections.

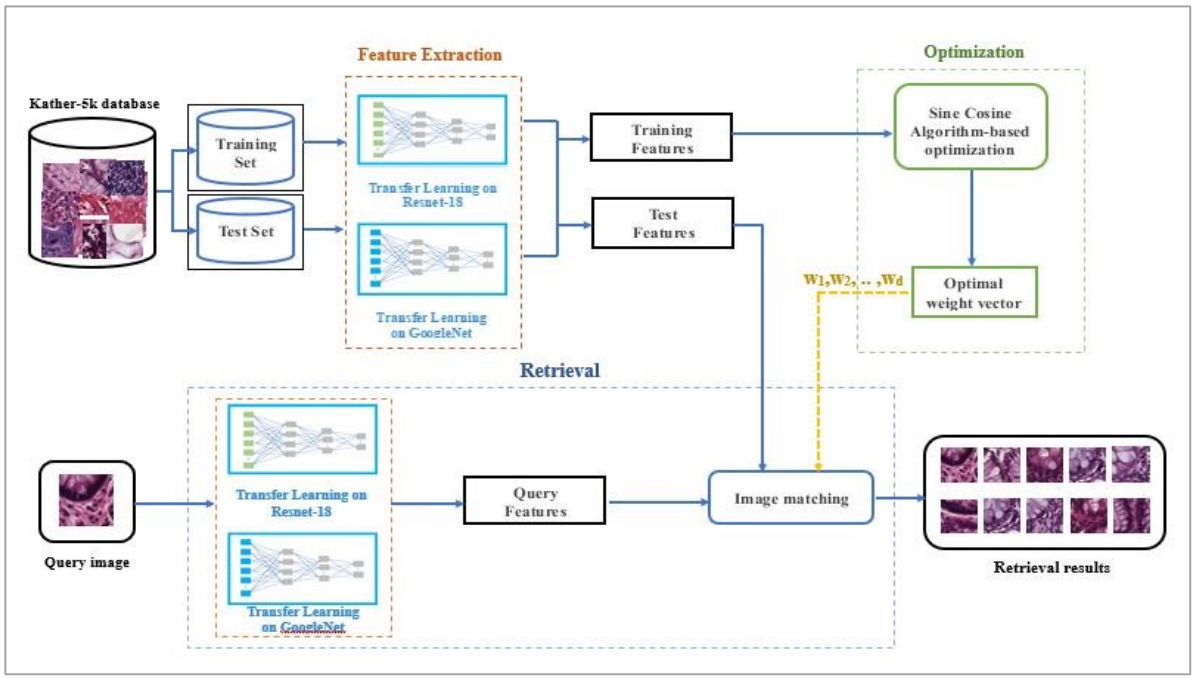


Figure 6. 4: Proposed CBHIR system for colon images (Boukerma et al, 2024)

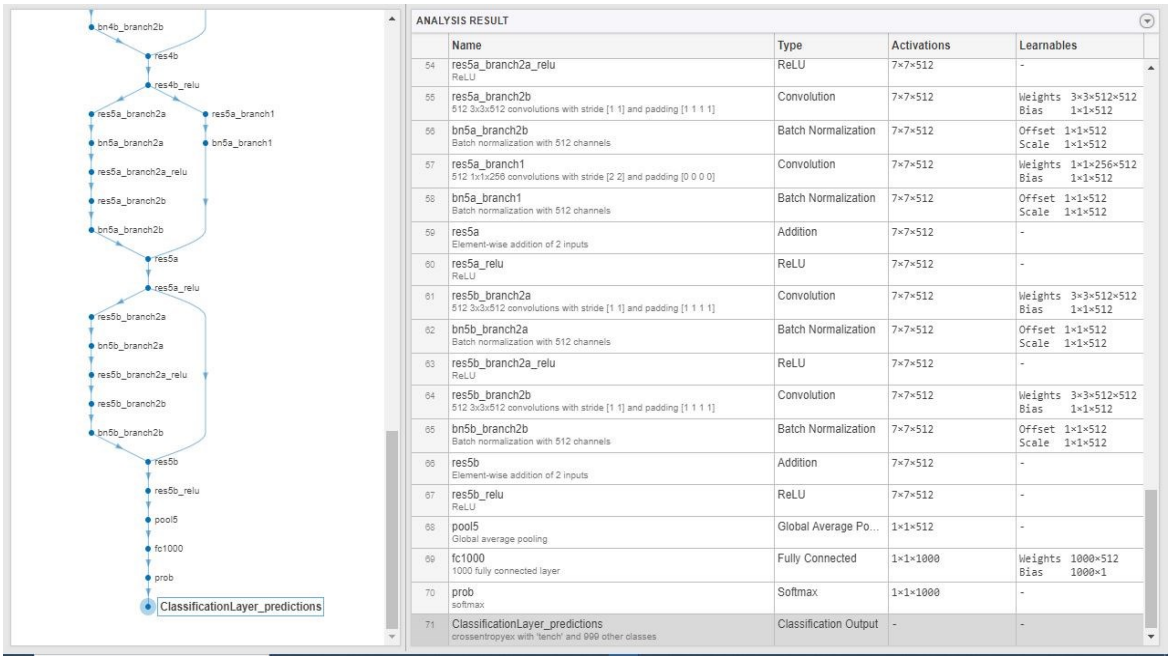
Feature learning: To learn high level features from colon histology images, the latter are first pre-processed. Thereafter, deep features are extracted using transfer learning then normalized.

1. *Preprocessing*: To enable the processing of the images by the two pre-trained CNNs, the Kather-5k images are resized to the input resolution of these CNNs. In particular, the Kather-5k images which have a resolution of 150 × 150 are scaled-up to the input size of ResNet18 (i.e., 224× 224) and GoogLeNet (i.e., 224× 224).

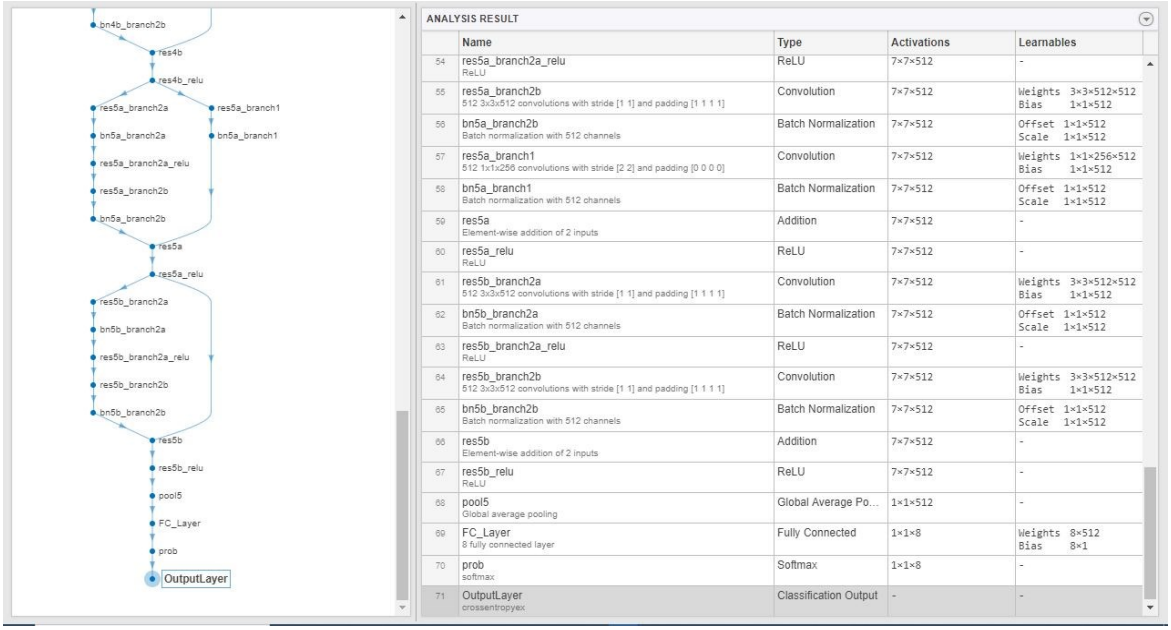
2. *Feature extraction*: As mentioned previously, we apply transfer learning on the two pre-trained CNN models, namely ResNet18 and GoogLeNet, in order to extract the deep features. The choice of these two CNN models is motivated by simplicity of their architectures, as well as the promising performance demonstrated by them in numerous works dealing with colon histology images (Al-Jabbar et al, 2023; Tsai & Tao, 2021). Thus, we first extract the features from each CNN model separately by applying transfer learning; then, we combine the extracted features in a single deep feature vector.

    - Transfer learning using ResNet18 (He et al, 2016): This CNN, which consists 18 layers, was trained on the ImageNet database made up of over one million

images with one thousand classes. To adapt the ResNet18 model to our dataset which contains 5000 images with eight classes, we apply transfer learning method by making few changes to the final layers and parameters of the model. Thus, after splitting the image dataset into training and testing sets, we apply the preprocessing described previously; then, we create two new layers to replace the last fully connected layer (fc1000) and the classification output layer (classificationLayer_Predictions). Thereafter, we fine-tune the model hyper-parameters as follows: Adam optimizer, a mini-batch size of 64, a max epoch number of 20 and an initial learning rate of 1e-4. Finally, we retrain and test the modified pre-trained ResNet18 model on Kather-5k dataset. We thus extract a feature vector of size 512 bins from the pool5 layer. The final layers of ResNet18 network before and after applying transfer learning, visualized using MATLAB's Deep Learning Toolbox, are illustrated in Figure 6.5.



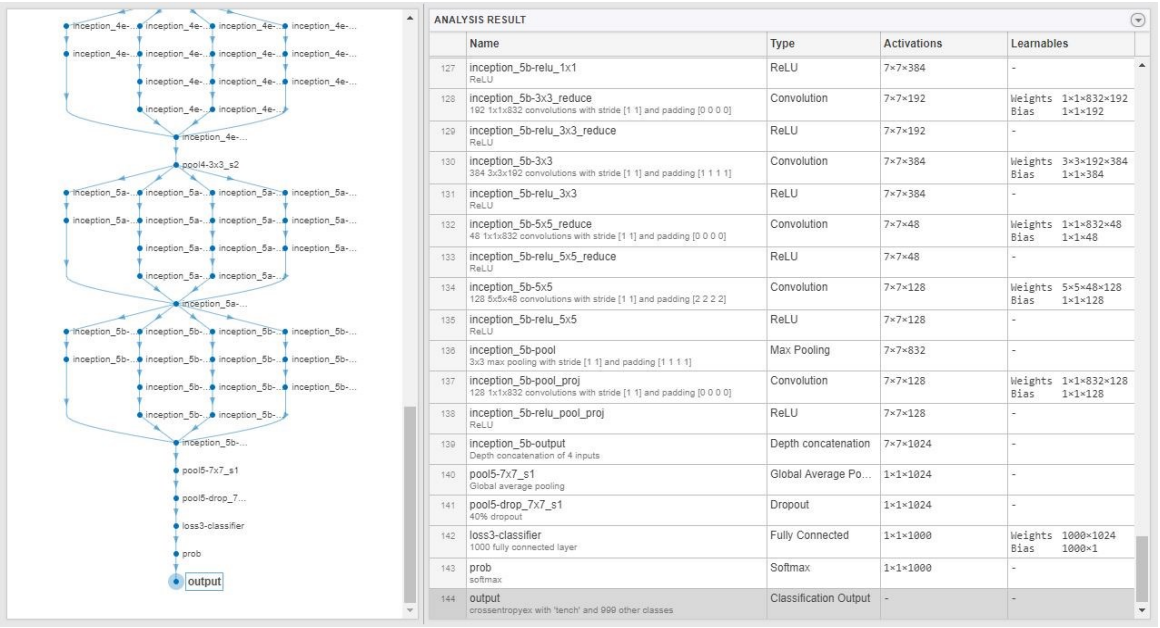
(a)



(b)

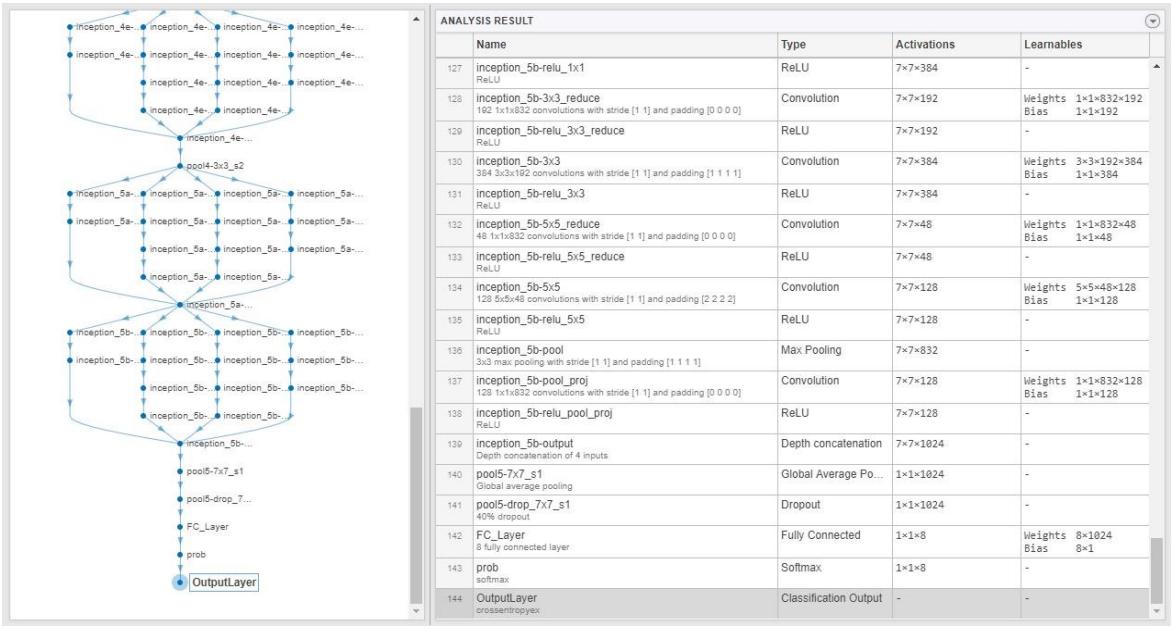
Figure 6. 5: Comparison of ResNet18 architecture before (a) and after (b) the application of transfer learning

- Transfer learning using GoogLeNet (Szegedy et al, 2015): Likewise, the same process is repeated by using the pre-trained GoogLeNet CNN. The latter is developed by researchers at Google, and consists of 22 layers. GoogLeNet was also trained on the ImageNet database for the classification task. The images in this database are grouped into 1,000 different classes. To adapt the GoogLeNet to Kather-5k dataset, we apply the same process used for ResNet18. The difference here is that the layers replaced are the fully connected layer (loss3-classifier) and the classification output layer (output). As for the model hyper-parameters, they are fine-tuned as follows: Adam optimizer, a mini-batch size of 10, a max epoch number of 20 and an initial learning rate of 1e-4. Then, the model is retrained and tested on Kather-5k dataset. Ultimately, 1024 features are extracted from the pool5-7x7_s1 layer of the transferred model. The final layers of GoogleNet network before and after applying transfer learning, visualized using MATLAB's Deep Learning Toolbox, are illustrated in Figure 6.6.



	Name	Type	Activations	Learnables
127	inception_5b-relu_1x1 ReLU	ReLU	7×7×384	-
128	inception_5b-3x3_reduce 192 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	7×7×192	Weights 1×1×832×192 Bias 1×1×192
129	inception_5b-relu_3x3_reduce ReLU	ReLU	7×7×192	-
130	inception_5b-3x3 384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	7×7×384	Weights 3×3×192×384 Bias 1×1×384
131	inception_5b-relu_3x3 ReLU	ReLU	7×7×384	-
132	inception_5b-5x5_reduce 48 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	7×7×48	Weights 1×1×832×48 Bias 1×1×48
133	inception_5b-relu_5x5_reduce ReLU	ReLU	7×7×48	-
134	inception_5b-5x5 128 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]	Convolution	7×7×128	Weights 5×5×48×128 Bias 1×1×128
135	inception_5b-relu_5x5 ReLU	ReLU	7×7×128	-
136	inception_5b-pool 3x3 max pooling with stride [1 1] and padding [1 1 1 1]	Max Pooling	7×7×832	-
137	inception_5b-pool_proj 128 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	7×7×128	Weights 1×1×832×128 Bias 1×1×128
138	inception_5b-relu_pool_proj ReLU	ReLU	7×7×128	-
139	inception_5b-output Depth concatenation of 4 inputs	Depth concatenation	7×7×1024	-
140	pool5-7x7_s1 Global average pooling	Global Average Po...	1×1×1024	-
141	pool5-drop_7x7_s1 40% dropout	Dropout	1×1×1024	-
142	loss3-classifier 1000 fully connected layer	Fully Connected	1×1×1000	Weights 1000×1024 Bias 1000×1
143	prob softmax	Softmax	1×1×1000	-
144	output crossentropyex with 'tench' and 999 other classes	Classification Output	-	-

(a)



	Name	Type	Activations	Learnables
127	inception_5b-relu_1x1 ReLU	ReLU	7×7×384	-
128	inception_5b-3x3_reduce 192 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	7×7×192	Weights 1×1×832×192 Bias 1×1×192
129	inception_5b-relu_3x3_reduce ReLU	ReLU	7×7×192	-
130	inception_5b-3x3 384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	7×7×384	Weights 3×3×192×384 Bias 1×1×384
131	inception_5b-relu_3x3 ReLU	ReLU	7×7×384	-
132	inception_5b-5x5_reduce 48 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	7×7×48	Weights 1×1×832×48 Bias 1×1×48
133	inception_5b-relu_5x5_reduce ReLU	ReLU	7×7×48	-
134	inception_5b-5x5 128 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]	Convolution	7×7×128	Weights 5×5×48×128 Bias 1×1×128
135	inception_5b-relu_5x5 ReLU	ReLU	7×7×128	-
136	inception_5b-pool 3x3 max pooling with stride [1 1] and padding [1 1 1 1]	Max Pooling	7×7×832	-
137	inception_5b-pool_proj 128 1x1x832 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	7×7×128	Weights 1×1×832×128 Bias 1×1×128
138	inception_5b-relu_pool_proj ReLU	ReLU	7×7×128	-
139	inception_5b-output Depth concatenation of 4 inputs	Depth concatenation	7×7×1024	-
140	pool5-7x7_s1 Global average pooling	Global Average Po...	1×1×1024	-
141	pool5-drop_7x7_s1 40% dropout	Dropout	1×1×1024	-
142	FC_Layer 8 fully connected layer	Fully Connected	1×1×8	Weights 8×1024 Bias 8×1
143	prob softmax	Softmax	1×1×8	-
144	OutputLayer crossentropyex	Classification Output	-	-

(b)

Figure 6. 6: Comparison of GoogleNet architecture before (a) and after (b) the application of transfer learning

3. Normalization: Before using the extracted features in the next phases, they are first combined then normalized. In particular, the features containing negative and positive values are normalized within the range [0, 1].

Optimization: To reduce the dimension of the combined deep feature vectors derived during the preceding phase, a set of optimal weights is generated by SCA algorithm for the selection

of the most discriminative bins from the combined features. In this context, we apply SCA on the optimization (training) set of the database while considering the average precision rate (the average recall rate or the average F-score) as a fitness function. In this phase which performs offline, the SCA starts by generating randomly the initial population in the range [0,0.1]. Then, the fitness function of each individual is computed and evaluated. The individual with best fitness is deemed the best solution, and its position as well as the SCA parameters are updated. The process continues by re-evaluating and updating the solutions until the maximum number of iterations is met.

Retrieval: This phase consists, in fact, in an online process that aims to search and display the images that are visually the most similar to the user query image. For this, the features extracted from the query image are usually compared to those extracted from all images in the test database. In this work, only a subset of the features bins is used for the comparison. More precisely, only the selected bins among the full feature vectors are used in the comparison. In particular, the optimal weight vector generated during the optimization phase is used for selecting only the bins that their corresponding weight values are greater than 0.05. The bins corresponding to the weight values below this threshold are thus removed. As in the previous works, the threshold is determined experimentally. Moreover, the selected bins are weighted by their corresponding weights to enhance the retrieval performance.

6.4.3 Experimental evaluation

In order to evaluate the influence of the optimized deep features extracted from the colon histology images on the CBHIR performance, various experiments were conducted on Kather-5k dataset. These experiments are described in this section, and their outcomes are reported and thoughtfully analyzed.

6.4.3.1 Experimental details

The proposed method was implemented using MATLAB R2020a, on a machine with an Intel® Core i7- 2.6 GHz processor and 8 GB internal RAM. For the optimization phase, the standard version of SCA was used, with following parameters: population size=20 individuals, maximum number of iterations=5. As for the image dataset, as above-mentioned, we used Kather-5k dataset, which was randomly divided into 70% training and 30% testing. The training set was used in the optimization phase, as well as in transfer learning to train the two CNNs models. To assess the performance of the proposed method, we used three

evaluation metrics, namely average precision rate (APR), average recall rate (ARR) and average F-score.

6.4.3.2 Results and analysis

A comparison of the image retrieval results, in terms of APR, ARR and average F-score, and the feature size between the proposed optimized deep features (reduce set features) and the non-optimized deep features (full set features) is provided in Table 6.8. These results were computed for the top 20 retrieved images using the Manhattan distance.

Table 6.8: Comparison of the feature dimension and performance for the top 20 images (Boukerma et al, 2024)

Method	Average precision rate		Average recall rate		Average F-score	
	No. of features	*APR (%)*	*No. of features*	*ARR (%)*	*No. of features*	*Avg. F-score (%)*
Combined features (without optimization)	1536	94.08	1536	10.06	1536	18.18
Proposed method (optimized features)	**763**	**94.10**	**797**	**10.06**	**785**	**18.18**

As shown in Table 6.8, the proposed optimized deep features show a significant reduction in the feature size compared to the combined features, for up to 50%, while still maintaining high APR, ARR and average F-score results. Specifically, when the APR was considered as a fitness function, the dimension of the features was reduced from 1536 to 763 bins, while keeping the highest APR with 94.10%. Likewise, when using the ARR as a fitness function, the features size was decreased from 1536 to 797 bins, while retaining almost the same ARR value. As for the average F-score, the optimized features maintained almost the same value (18.18%) as when using the full set features, however, the dimension was reduced from 1536 for the full set features (combined features) to 785 bins for the optimized features, with a reduction rate of 49%.

In fact, reducing the dimension of the feature space up to 50% leads consequently to a decrease in the amount of memory space used by the CBHIR system. Furthermore, it is also important to reduce the time complexity of the proposed system. To assess the time complexity of our system, we calculate the average retrieval time of all query images in the test database while varying the number of retrieved images. Table 6.9 presents a comparison of the average retrieval time between the proposed method, i.e. using the optimized deep

features, and the combined features (without optimization), according to different top-k retrieved images. These results were computed using Manhattan distance.

Table 6.9**:** Comparative table of the average retrieval time (seconds) (Boukerma et al, 2024)

Method	*Top 5*	*Top 10*	*Top 20*	*Top 30*
Combined features (without optimization)	0.0250	0.0312	0.0348	0.0357
Proposed method (optimized features)	**0.0166**	**0.0171**	**0.0190**	**0.0221**

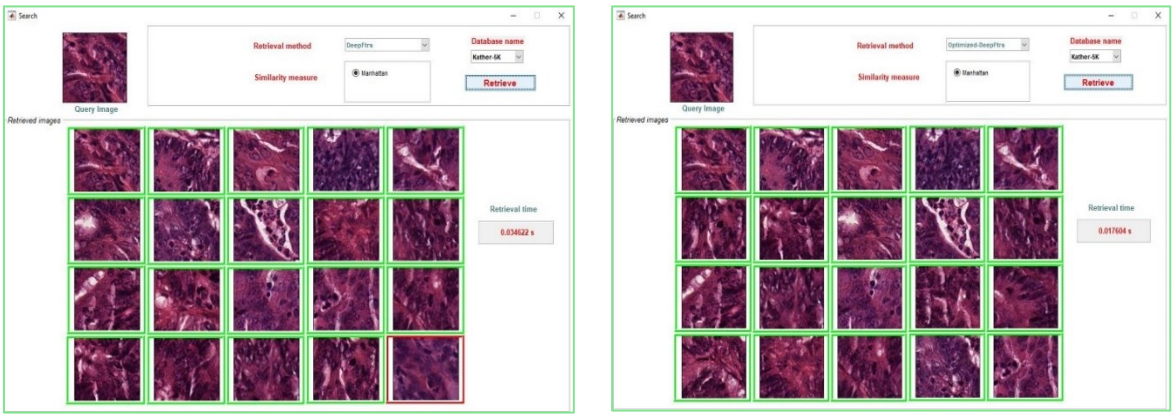
According to Table 6.9, the average retrieval time taken when using the proposed optimized deep features is significantly shorter than that taken when using the non-optimized deep features. For instance, for the top 5 retrieved images, the average retrieval time was decreased from 0.0250 sec to 0.0166 sec, and from 0.0348 sec to 0.0190 sec for the top 20 retrieved images.

Table 6.10 provides a comparison of the APR results between the proposed method and some state-of-the-art methods, including deep learning- and handcrafted-based methods. These results were computed for different retrievals (top 5, top 10, top 20 and top 30), using the Manhattan distance.

Table 6**.**10: Comparison of the average precision rates between the proposed and some state-of-the-art methods (Boukerma et al, 2024)

Method	*Top 5*	*Top 10*	*Top 20*	*Top 30*
ResNet18 (He et al, 2016)	95.13	94.04	92.55	91.51
GoogLeNet (Szegedy et al, 2015)	95.59	94.59	93.86	93.35
MobileNetV2 (Sandler et al, 2018)	94.14	92.75	91.35	90.43
Riesz+M-LBP (Yazdi & Erfankhah, 2020)	86.73	83.60	79.72	77.00
Proposed method	**96.04**	**94.97**	**94.10**	**93.52**

Figure 6.7 depicts an example of a query image from the TUMOR class of Kather-5k database. Figure 6.7 (a) illustrates the 20 retrieved images displayed to the user when the full set of the deep features was used. Figure 6.7 (b) illustrates the 20 retrieved images for the same query image, but when using the optimized deep features.



(a) (b)

Figure 6. 7: Example of a query images from Kather-5k database, using: (a) the full set deep features, (b) the optimized deep features (Boukerma et al, 2024)

From Figure 6.7, it is observed that the precision of the query image when using the proposed optimized deep features was increased from 19/20 to 20/20, while the retrieval time was decreased from 3.4622e-2 sec to 1.7604e-2 sec.

6.5 Conclusion

In this chapter, three feature dimensionality reduction methods were proposed. The first two methods were designed to reduce the features extracted through local patterns descriptors, while the third method was specifically developed for reducing the features extracted using deep learning-based methods. All the proposed methods were particularly designed to deal with histopathology images, and served as a tool for enhancing the automatic analysis of this type of images and therefore facilitate the disease diagnosis. While the first dimensionality method was based on PSO algorithm, the second and third were based on SCA algorithm. On the other hand, in the first and second contributions the features were extracted using color local pattern methods, specifically, color ILBP for the first contribution, and color LBP and OCLBP for the second contribution. In the third contribution, however, the features were extracted using CNN models, namely ResNet18 and GoogLeNet. By enhancing the CBHIR effectiveness and efficiency, it may be concluded that all the proposed methods successfully met the initially defined objectives.

General conclusions and perspectives

Conclusions

Despite the ongoing success of CBIR systems, they still face many open and unresolved challenges, including the semantic gap, the curse of dimensionality and the effectiveness of the feature representations extracted from images. Accordingly, this thesis was conducted with the aim of developing effective and efficient methods able to address the key challenges of CBIR. Specifically, our study was directed toward achieving two main objectives: developing robust feature representations able to improve the effectiveness of CBIR, thereby reducing the semantic gap; and suggesting novel feature reduction methods capable of enhancing the CBIR efficiency. To accomplish the first objective, we proposed the development of novel feature descriptors for extracting discriminative feature representations, as well as the optimization of some existing feature extraction methods using optimization and learning techniques. In particular, we prioritized local pattern methods for feature extraction owing to their strength in capturing the texture information, and especially in identifying and detecting the micro patterns in images. This capability is of great importance for many types of images, especially for some forms of biomedical images, such as histopathology images. To achieve the second objective, the capabilities of optimization and learning techniques were investigated in order to optimize the features extracted from images and reduce their dimension. In particular, we proposed three feature dimensionality reduction methods based on some metaheuristic algorithms to reduce the size of the features extracted using either local pattern methods or pre-trained CNN models.

The objectives and motivations behind this study, as well as the contributions of the thesis, were clearly stated in the introductory chapter. Following this, the fundamental concepts related to the CBIR area was presented in Chapter2. In particular, the chapter outlined the progression of image retrieval systems, with particular attention to CBIR systems and their fundamental components. Moreover, a greater emphasis was given to the feature extraction component as it has a major impact on the CBIR performance. Chapter 3 was particularly devoted to the local pattern methods, highlighting the LBP method and its color variants with their strengths and limitations. The different techniques used for optimizing the local pattern

methods, specifically pattern selection and pattern weighting, were also investigated in this chapter. In Chapter 4, an in-depth review of optimization and learning techniques was provided. In particular, a structured taxonomy of the existing optimization and machine learning techniques was outlined, focusing specifically on those used in the context of CBIR.

After presenting the literature review, the subsequent two chapters were devoted to the main contributions of this study. Chapter 5 was particularly dedicated to the first two contributions, which involve the development of discriminative and robust feature representations, aiming at addressing the semantic gap issue and improving the CBIR effectiveness. In the first contribution, two color local pattern descriptors were proposed, namely O-OCLBP and O-IOCLBP. In fact, by enhancing the robustness against illumination change of the existing OCLBP and IOCLBP methods, the proposed O-OCLBP and O-IOCLBP methods could achieve significant improvements in the retrieval performance. However, it should be noted that the two proposed methods remain constrained by the high dimension of the features derived using O-OCLBP and O-IOCLBP descriptors. Nonetheless, this limitation does not diminish the relevance and effectiveness of the proposed methods, but it addresses a critical need for further refinements. In the second contribution, an effective weighting mechanism based on DE algorithm was introduced to enhance the performance of some exciting grayscale local pattern methods, namely LBP, ILBP, LTP and MLBP. The proposed dynamic pattern weighting method showed clear improvement in the retrieval performance over the existing methods, demonstrating its effectiveness in enhancing the image representation. However, the retrieval performance of the proposed dynamic weighted local pattern methods, like most grayscale LBP variants, remain lower when used alone. Building on the observations and limitations discussed in Chapter 5, three other contributions were introduced in Chapter 6. In fact, the contributions presented in Chapter 6 specifically target the curse of dimensionality, aiming to reduce the high-dimensional features extracted from histopathological images. In particular, three feature dimensionality reduction methods were proposed. In the third contribution of this thesis, a significant feature dimensionality reduction method based on PSO algorithm was proposed in order to reduce the dimension of some combined local pattern features extracted in multiple color spaces. This method was further expanded in the fourth contribution by using a relatively recent and computationally efficient metaheuristic, which is SCA algorithm, in addition to other local pattern methods and more histopathology image datasets. Moreover, SCA algorithm was used in the fifth

contribution to select the most relevant bins from some combined deep features extracted using ResNet18 and GoogLeNet CNN models, and therefore reducing their dimension.

Finally, we consider that the methods proposed in this thesis successfully addressed the stated objectives and provided validated solutions to the identified CBIR issues.

Perspectives

Based on the outcomes of this thesis, several perspectives for future work can be envisioned.

- One promising direction is to improve the efficiency of the proposed O-OCLBP and O-IOCLBP descriptors by applying one of the proposed dimensionality reduction methods proposed in this research.
- An interesting extension of the proposed opponent color scheme would involve its application on other color LBP variants to derive more robust and effective descriptors.
- Further investigations could explore other deep learning techniques that enable extracting more discriminative and richer semantic features from more deeper layers.
- Additionally, applying our proposed color-texture descriptors to others domains with strong local texture information, such as Texture classification, Face recognition and Biometric applications, constitutes a promising direction for broadening the impact and applicability of the proposed descriptors.

Bibliography

Abdesselam, A. (2009). "Texture image retrieval using Fourier transform", in *Proc. Int. Conf. Commun., Comput. Power (ICCCP'09)*. 1-6.

Abduljabbar, Z. A., Ibrahim, A., Hussain, M. A., Hussien, Z. A., Al Sibahee, M. A. & Lu, S. (2019). "EEIRI: efficient encrypted image retrieval in IoT-cloud", *KSII Transactions on Internet and Information Systems (TIIS)*, 13(11), 5692-5716.

Aboaisha, H. (2015). "*The Optimisation of Elementary and Integrative Content-Based Image Retrieval Techniques*" University of Huddersfield.

Adegbola, O. A., Adeyemo, I. A., Semire, F. A., Popoola, S. I. & Atayero, A. A. (2020). "A principal component analysis-based feature dimensionality reduction scheme for content-based image retrieval system", *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 18(4), 1892-1896.

Admile, N. S., Jadhav, A. A., Karve, S. M. & Kasture, A. A. (2021). "Content-based image retrieval using color histogram and bit pattern features", in *Techno-Societal 2020: Proceedings of the 3rd International Conference on Advanced Technologies for Societal Applications—Volume 1*. Springer, 499-508.

Agarwal, M. (2023). "Image retrieval system using kirsch based local ternary pattern", *Advances in Electrical and Electronic Engineering*, 21(1), 28.

Agarwal, M., Singhal, A. & Lall, B. (2019). "Multi-channel local ternary pattern for content-based image retrieval", *Pattern Analysis and Applications*, 22(4), 1585-1596.

Aher, P. & Lilhore, U. (2016). "An improved cbmir architecture, based on modified classifiers & feedback method for tumor image retrieval from mri images", *International Journal of Modern Trends in Engineering and Research (IJMTER) Volume*, 3(12), 156-160.

Ahmed, A. S. & Ibraheem, I. N. (2024). "Enhanced Low-Level-Feature-and Color-Aware Content-Based Image Retrieval Using Deep Learning", *International Journal of Intelligent Engineering & Systems*, 17(6), 436-449.

Ahmed, F., Kabir, M. H., Bhuyan, S., Bari, H. & Hossain, E. (2014). "Automated weed classification with local pattern-based texture descriptors", *Int. Arab J. Inf. Technol.*, 11(1), 87-94.

Ahmed, K. T., Jaffar, S., Hussain, M. G., Fareed, S., Mehmood, A. & Choi, G. S. (2021). "Maximum response deep learning using Markov, retinal & primitive patch binding with GoogLeNet & VGG-19 for large image retrieval", *Ieee Access*, 9, 41934-41957.

Akçiçek, M., Karaduman, M., Petik, B., Ünlü, S., Mutlu, H. B. & Yildirim, M. (2025). "Detection of Acromion Types in Shoulder Magnetic Resonance Image Examination with Developed Convolutional Neural Network and Textural-Based Content-Based Image Retrieval System", *Journal of Clinical Medicine*, 14(2), 505.

Al-Abaji, M. A. (2019). "Cuckoo Search Algorithm Based Feature Selection in Image Retrieval System", *Journal of Education and Practice*, 10(15), 58-65.

Al-Jabbar, M., Alshahrani, M., Senan, E. M. & Ahmed, I. A. (2023). "Histopathological Analysis for Detecting Lung and Colon Cancer Malignancies Using Hybrid Systems with Fused Features", *Bioengineering*, 10(3), 383.

Alatas, B. (2011). "ACROA: artificial chemical reaction optimization algorithm for global optimization", *Expert Systems with Applications*, 38(10), 13170-13180.

Alizadeh, S. M., Helfroush, M. S. & Müller, H. (2023). "A novel Siamese deep hashing model for histopathology image retrieval", *Expert Systems with Applications*, 225, 120169.

Alkhawlani, M., Elmogy, M. & El Bakry, H. (2015). "Text-based, content-based, and semantic-based image retrievals: a survey", *Int. J. Comput. Inf. Technol*, 4(01), 58-66.

Alrahhal, M. & Supreethi, K. (2024a). "Enhancing image retrieval accuracy through multi-resolution HSV-LNP feature fusion and modified K-NN relevance feedback", *International Journal of Information Technology*, 1-15.

Alrahhal, M. & Supreethi, K. (2024b). "Integrating machine learning algorithms for robust content-based image retrieval", *International Journal of Information Technology*, 16(8), 5005-5021.

Alsmadi, M. K. (2017). "An efficient similarity measure for content based image retrieval using memetic algorithm", *Egyptian journal of basic and applied sciences*, 4(2), 112-122.

Alzu'bi, A., Amira, A. & Ramzan, N. (2015). "Semantic content-based image retrieval: A comprehensive study", *Journal of Visual Communication and Image Representation*, 32, 20-54.

Andrysiak, T. & Choraś, M. (2005). "Image retrieval based on hierarchical Gabor filters".

Arebey, M., Hannan, M., Begum, R. & Basri, H. (2011). "CBIR for an automated solid waste bin level detection system using GLCM", in *Visual Informatics: Sustaining Research and Innovations: Second International Visual Informatics Conference, IVIC 2011, Selangor, Malaysia, November 9-11, 2011, Proceedings, Part I 2*. Springer, 280-288.

Arevalillo-Herráez, M., Domingo, J. & Ferri, F. J. (2008). "Combining similarity measures in content-based image retrieval", *Pattern recognition letters*, 29(16), 2174-2181.

Arora, N., Kakde, A. & Sharma, S. C. (2023). "An optimal approach for content-based image retrieval using deep learning on COVID-19 and pneumonia X-ray Images", *International Journal of System Assurance Engineering and Management*, 14(Suppl 1), 246-255.

Arora, N., Sharma, P., Kumar, P. & Sharma, S. C. (2024). "RTLBP-AN Efficient Local Pattern For Facial Images Retrieval", in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3525-3529.

Arora, N. & Sharma, S. C. (2024). "Resource-Efficient Image Retrieval: A Study of Local Patterns Versus Deep Learning Models", in *International Conference on Machine Learning, Advances in Computing, Renewable Energy and Communication*. Springer, 57-66.

Arunkumar, N. & Ram, A. R. (2020). "Cbir systems: Techniques and challenges", in *2020 International conference on communication and signal processing (ICCSP)*. IEEE, 0141-0146.

Atashpaz-Gargari, E. & Lucas, C. (2007). "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition", in *2007 IEEE congress on evolutionary computation*. Ieee, 4661-4667.

Ayyachamy, S. & Manivannan, V. S. (2013). "Medical image registration-based retrieval using distance metrics", *International journal of imaging systems and technology*, 23(4), 360-371.

Ayyad, S. M., Shehata, M., Shalaby, A., Abou El-Ghar, M., Ghazal, M., El-Melegy, M., Abdel-Hamid, N. B., Labib, L. M., Ali, H. A. & El-Baz, A. (2021). "Role of AI and histopathological images in detecting prostate cancer: a survey", *Sensors*, 21(8), 2586.

Babu, E. K., Mistry, K., Anwar, M. N. & Zhang, L. (2022). "Facial feature extraction using a symmetric inline matrix-LBP variant for emotion recognition", *Sensors*, 22(22), 8635.

Backes, A. R., Casanova, D. & Bruno, O. M. (2012). "Color texture analysis based on fractal descriptors", *Pattern Recognition*, 45(5), 1984-1992.

Backes, A. R., Gerhardinger, L. C., Neto, J. d. E. S. B. & Bruno, O. M. (2015). "Medical image retrieval and analysis by Markov random fields and multi-scale fractal dimension", *Physics in Medicine & Biology*, 60(3), 1125.

Bama, B. S. & Raju, S. (2010). "Fourier based rotation invariant texture features for content based image retrieval", in *2010 National Conference on Communications (NCC)*. IEEE, 1-5.

Barrera-García, J., Cisternas-Caneo, F., Crawford, B., Gómez Sánchez, M. & Soto, R. (2023). "Feature selection problem and metaheuristics: a systematic literature review about its formulation, evaluation and applications", *Biomimetics*, 9(1), 9.

Basturk, B. (2006). "An artificial bee colony (ABC) algorithm for numeric function optimization", in *IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 2006*. 2006, 12.

Behnam, M. & Pourghassem, H. (2013). "Feature descriptor optimization in medical image retrieval based on genetic algorithm", in *2013 20th Iranian conference on biomedical engineering (ICBME)*. IEEE, 280-285.

Belarbi, M. A., Mahmoudi, S. & Belalem, G. (2017). "PCA as dimensionality reduction for large-scale image retrieval systems", *International Journal of Ambient Computing and Intelligence (IJACI)*, 8(4), 45-58.

Belarbi, M. A., Mahmoudi, S., Belalem, G., Mahmoudi, S. A. & Cools, A. (2022). "A New Comparative Study of Dimensionality Reduction Methods in Large-Scale Image Retrieval", *Big Data and Cognitive Computing*, 6(2), 54.

Belattar, K. & Mostefai, S. (2013). "CBIR with RF: Which technique for which image", in *2013 3rd International Symposium ISKO-Maghreb*. IEEE, 1-7.

Benloucif, S. & Boucheham, B. (2014). "Impact of feature selection on the performance of content-based image retrieval (CBIR)", in *2014 4th International Symposium ISKO-Maghreb: Concepts and Tools for knowledge Management (ISKO-Maghreb)*. IEEE, 1-7.

Bhattacharyya, D., Hazra, D. & Kim, T.-h. (2015). "Image Retrieval Process Based on Relevance Feedback and Ontology Using Decision Tree", *International Journal of Multimedia and Ubiquitous Engineering*, 10(10), 83-90.

Bhowmick, C., Dutta, P. K. & Mahadevappa, M. (2020). "Wavelet Transform and Texton based Analysis for Detection of Benign and Malignant Masses", in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2178-2181.

Bi, X. & Pan, T. (2017). "Relevance feedback image retrieval based on teaching-learning-based optimization algorithm", *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, 45(7), 1668-1676.

Bianconi, F., Bello-Cerezo, R. & Napoletano, P. (2017). "Improved opponent color local binary patterns: an effective local image descriptor for color texture classification", *Journal of Electronic Imaging*, 27(1), 011002.

Bougueroua, S. & Boucheham, B. (2014). "Ellipse based local binary pattern for color image retrieval", in *2014 4th International Symposium ISKO-Maghreb: Concepts and Tools for knowledge Management (ISKO-Maghreb)*. IEEE, 1-8.

Bougueroua, S. & Boucheham, B. (2015). "GLI-Color: Gradual Locality Integration of Color features for image retrieval", in *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*. 1-5.

Bougueroua, S. & Boucheham, B. (2018). "GLIBP: gradual locality integration of binary patterns for scene images retrieval", *Journal of Information Processing Systems*, 14(2), 469-486.

Boukerma, R., Boucheham, B. & Bougueroua, S. (2022). "Image Retrieval Based on Dynamic Weighted Patterns", in *2022 2nd International Conference on New Technologies of Information and Communication (NTIC)*. IEEE, 1-6.

Boukerma, R., Boucheham, B. & Bougueroua, S. (2023). "Significant Feature Dimensionality Reduction for Histopathology Image Retrieval as a Tool for Healthcare Decisions Support", in *2023 International Conference on Decision Aid Sciences and Applications (DASA)*. Annaba, Algeria: IEEE, 349-353.

Boukerma, R., Boucheham, B. & Bougueroua, S. (2024). "Optimized Deep Features for Colon Histology Image Retrieval", in *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. IEEE, 1-8.

Boukerma, R., Boucheham, B. & Bougueroua, S. (2025a). "Orthogonal opponent colour local binary patterns: a new colour-texture descriptor for content based-image retrieval", *International Journal of Computational Vision and Robotics*, 15(3), 351-378.

Boukerma, R., Boucheham, B. & Bougueroua, S. (2025b). "A Pattern selection Scheme for Content-Based Histopathological Image Retrieval: an Effective Tool for Feature Dimensionality Reduction ", *International Journal of Bioinformatics Research and Applications*.

Boukerma, R., Bougueroua, S. & Boucheham, B. (2019). "A Local Patterns Weighting Approach for Optimizing Content-Based Image Retrieval Using a Differential Evolution Algorithm", in *2019 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS)*. IEEE, 1, 1-8.

Breiman, L., Friedman, J., Olshen, R. A. & Stone, C. J. (1984). "*Classification and regression trees*", Belmont, CA: Wadsworth International.

Brodatz, P. (1966). "*Textures: a photographic album for artists and designers*". New York, USA: Dover Publications.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. & Shah, R. (1993). "Signature verification using a" siamese" time delay neural network", *Advances in neural information processing systems*, 6.

Cai, J., Liu, M., Zhang, Q., Shao, Z., Zhou, J., Guo, Y., Liu, J., Wang, X., Zhang, B. & Li, X. (2022). "Renal Cancer Detection: Fusing Deep and Texture Features from Histopathology Images", *BioMed Research International*, 2022, 1-17.

Camlica, Z., Tizhoosh, H. R. & Khalvati, F. (2015). "Medical image classification via SVM using LBP features from saliency-based folded data", in *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE, 128-132.

Caputo, B., Hayman, E. & Mallikarjuna, P. (2005). "Class-specific material categorisation", in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. IEEE, 2, 1597-1604.

Carneiro, G., Chan, A. B., Moreno, P. J. & Vasconcelos, N. (2007). "Supervised learning of semantic classes for image annotation and retrieval", *IEEE transactions on pattern analysis and machine intelligence*, 29(3), 394-410.

Carson, C., Belongie, S., Greenspan, H. & Malik, J. (2002). "Blobworld: Image segmentation using expectation-maximization and its application to image querying", *IEEE Transactions on pattern analysis and machine intelligence*, 24(8), 1026-1038.

Chang, S.-K., Jungert, E. & Li, Y. (1989). "Representation and retrieval of symbolic pictures using generalized 2D strings", in *Visual Communications and Image Processing IV*. SPIE, 1199, 1360-1372.

Chang, S.-K., Shi, Q.-Y. & Yan, C.-W. (1987). "Iconic indexing by 2-D strings", *IEEE Transactions on Pattern Analysis and Machine Intelligence* (3), 413-428.

Chavda, S. & Goyani, M. (2020). "Hybrid approach to content-based image retrieval using modified multi-scale LBP and color features", *SN Computer Science*, 1(6), 305.

Chehrehgosha, A. & Emadi, M. (2016). "Face detection using fusion of LBP and AdaBoost", *Journal of Soft Computing and Applications*, 2016(1), 1-10.

Chowdhury, G. G. (2010). "*Introduction to modern information retrieval*", Facet publishing.

Chuctaya, H., Portugal, C., Beltrán, C., Gutiérrez, J., López, C. & Túpac, Y. (2011). "M-CBIR: A medical content-based image retrieval system using metric data-structures", in *2011 30th International Conference of the Chilean Computer Science Society*. IEEE, 135-141.

Cortes, C. & Vapnik, V. (1995). "Support-vector networks", *Machine learning*, 20, 273-297.

Cox, I. J., Miller, M. L., Minka, T. P., Papathomas, T. V. & Yianilos, P. N. (2000). "The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments", *IEEE transactions on image processing*, 9(1), 20-37.

Dai, C., Zhu, Y. & Chen, W. (2006). "Seeker optimization algorithm", in *International conference on computational and information science*. Springer, 167-176.

Dalal, N. & Triggs, B. (2005). "Histograms of oriented gradients for human detection", in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Ieee, 1, 886-893.

Das, P. & Neelima, A. (2017). "An overview of approaches for content-based medical image retrieval", *International journal of multimedia information retrieval*, 6(4), 271-280.

Deole, P. A. & Longadge, R. (2014). "Content based image retrieval using color feature extraction with KNN classification", *IJCSMC*, 3(5), 1274-1280.

Deselaers, T., Keysers, D. & Ney, H. (2005). "Fire–flexible image retrieval engine: Imageclef 2004 evaluation", in *Multilingual Information Access for Text, Speech and Images: 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004, Bath, UK, September 15-17, 2004, Revised Selected Papers 5*. Springer, 688-698.

Dey, S., Dutta, A., Ghosh, S. K., Valveny, E., Lladós, J. & Pal, U. (2019). "Aligning salient objects to queries: a multi-modal and multi-object image retrieval framework", in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part II 14*. Springer, 241-255.

Di Mascio, T., Frigioni, D. & Tarantino, L. (2010). "VISTO: A new CBIR system for vector images", *Information Systems*, 35(7), 709-734.

Dixit, A. & Hegde, N. P. (2013). "Image texture analysis-survey", in *2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT)*. IEEE, 69-76.

Dorigo, M., Birattari, M. & Stutzle, T. (2007). "Ant colony optimization", *IEEE computational intelligence magazine*, 1(4), 28-39.

Dou, Z., Wang, Z., Chen, W., Li, Y. & Wang, S. (2022). "Reliability-aware prediction via uncertainty learning for person image retrieval", in *European Conference on Computer Vision*. Springer, 588-605.

Duan, J. & Kuo, C.-C. J. (2022). "Bridging gap between image pixels and semantics via supervision: a survey", *APSIPA Transactions on Signal and Information Processing*, 11(1), 1-45.

Dubey, R. S., Choubey, R. & Bhattacharjee, J. (2010). "Multi feature content based image retrieval", *International Journal on Computer Science and Engineering*, 2(6), 2145-2149.

Dubey, S. R. & Mukherjee, S. (2020). "Ldop: local directional order pattern for robust face retrieval", *Multimedia Tools and Applications*, 79(9), 6363-6382.

Dubey, S. R., Singh, S. K. & Singh, R. K. (2016). "Multichannel decoded local binary patterns for content-based image retrieval", *IEEE transactions on image processing*, 25(9), 4018-4032.

Duygulu, P., Barnard, K., de Freitas, J. F. & Forsyth, D. A. (2002). "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary", in *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part IV 7*. Springer, 97-112.

Eakins, J. (2002). "Content-based image retrieval–what's holding it back", in *ASCI [Advanced School for Computing and Imaging] 2002 Conference, Lochem, the Netherlands*.

El-Naqa, I., Yang, Y., Galatsanos, N. P., Nishikawa, R. M. & Wernick, M. N. (2004). "A similarity learning approach to content-based image retrieval: application to digital mammography", *IEEE transactions on medical imaging*, 23(10), 1233-1244.

Elias, S. J., Hatim, S. M., Hassan, N. A., Abd Latif, L. M., Ahmad, R. B., Darus, M. Y. & Shahuddin, A. Z. (2019). "Face recognition attendance system using Local Binary Pattern (LBP)", *Bulletin of Electrical Engineering and Informatics*, 8(1), 239-245.

Enser, P. (2008). "The evolution of visual information retrieval", *Journal of Information Science*, 34(4), 531-546.

Escalante, H. J., Hernández, C., López, A., Marín, H., Montes, M., Morales, E., Sucar, E. & Villasenor, L. (2008). "Towards annotation-based query and document expansion for image retrieval", in *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers 8*. Springer, 546-553.

Fadaei, S., Amirfattahi, R. & Ahmadzadeh, M. R. (2017). "New content-based image retrieval system based on optimised integration of DCD, wavelet and curvelet features", *IET Image Processing*, 11(2), 89-98.

Fei-Fei, L., Fergus, R. & Perona, P. (2004). "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories", in *2004 conference on computer vision and pattern recognition workshop*. IEEE, 178-178.

Fei-Fei, L. & Perona, P. (2005). "A bayesian hierarchical model for learning natural scene categories", in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. IEEE, 2, 524-531.

Fisher, R. A. (1936). "The use of multiple measurements in taxonomic problems", *Annals of eugenics*, 7(2), 179-188.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D. & Petkovic, D. (1995). "Query by image and video content: The QBIC system", *computer*, 28(9), 23-32.

Gain, A. (2024). "Optimization of CNN for Content-Based Image Retrieval in Healthcare", *Internet of Things-Based Machine Learning in Healthcare*Chapman and Hall/CRC, 96-125.

Galshetwar, G. M., Waghmare, L. M., Gonde, A. B. & Murala, S. (2017). "Edgy salient local binary patterns in inter-plane relationship for image retrieval in diabetic retinopathy", *Procedia computer science*, 115, 440-447.

Ganasala, P. & Prasad, A. D. (2020). "Medical image fusion based on laws of texture energy measures in stationary wavelet transform domain", *International Journal of Imaging Systems and Technology*, 30(3), 544-557.

Gao, J., Wang, Z. & Xia, J. (2021). "Sparse Representation Algorithm Based on Block LBP and Adaptive Weighting", in *Proceedings of 2021 Chinese Intelligent Systems Conference: Volume I*. Springer, 163-173.

Gautam, G. & Khanna, A. (2024). "Content Based Image Retrieval System Using CNN based Deep Learning Models", *Procedia Computer Science*, 235, 3131-3141.

Georgioudakis, M. & Plevris, V. (2020). "A comparative study of differential evolution variants in constrained structural optimization", *Frontiers in Built Environment*, 6, 102.

Ghahremani, M., Ghadiri, H. & Hamghalam, M. (2021). "Local features integration for content-based image retrieval based on color, texture, and shape", *Multimedia Tools and Applications*, 80(18), 28245-28263.

Ghorbani, N. & Babaei, E. (2014). "Exchange market algorithm", *Applied soft computing*, 19, 177-187.

Glover, F. (1986). "Future paths for integer programming and links to artificial intelligence", *Computers & operations research*, 13(5), 533-549.

Gravina, M., Marrone, S., Piantadosi, G., Moscato, V. & Sansone, C. (2021). "Developing a smart PACS: CBIR system using deep learning", in *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part II*. Springer, 296-309.

Griffin, G., Holub, A. & Perona, P. (2007). *"Caltech-256 object category dataset*".

Grycuk, R., Gabryel, M., Nowicki, R. & Scherer, R. (2016). "Content-based image retrieval optimization by differential evolution", in *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 86-93.

Gudivada, V. N. & Raghavan, V. V. (1995). "Design and evaluation of algorithms for image retrieval by spatial similarity", *ACM Transactions on Information Systems (TOIS)*, 13(2), 115-144.

Guo, Z., Zhang, L. & Zhang, D. (2010). "A completed modeling of local binary pattern operator for texture classification", *IEEE transactions on image processing*, 19(6), 1657-1663.

Gupta, M., Bhatnagar, C. & Jalal, A. (2018). "Clothing image retrieval based on multiple features for smarter shopping", *Procedia Computer Science*, 125, 143-148.

Gurubelli, Y., Ramanathan, M. & Ponnusamy, P. (2023). "Colour texture descriptor for CBIR of diseased tomato leaf images using modified local zigzag pattern", *Multimedia Tools and Applications*, 82(24), 38077-38095.

Hadid, M., Hussein, Q. M., Al-Qaysi, Z., Ahmed, M. & Salih, M. M. (2023). "An Overview of Content-BasedImage Retrieval Methods And Techniques", *Iraqi Journal For Computer Science and Mathematics*, 4(3), 6.

Haji, M. S., Alkawaz, M. H., Rehman, A. & Saba, T. (2019). "Content-based image retrieval: A deep look at features prospectus", *International Journal of Computational Vision and Robotics*, 9(1), 14-38.

Hameed, I. M., Abdulhussain, S. H. & Mahmmod, B. M. (2021). "Content-based image retrieval: A review of recent trends", *Cogent Engineering*, 8(1), 1927469.

Hanif, M. A., Kaur, H., Rakhra, M. & Singh, A. (2022). "Role of CBIR In a Different fields-An Empirical Review", in *2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST)*. IEEE, 1-7.

Haralick, R. M. (1979). "Statistical and structural approaches to texture", *Proceedings of the IEEE*, 67(5), 786-804.

Haralick, R. M., Shanmugam, K. & Dinstein, I. H. (1973). "Textural features for image classification", *IEEE Transactions on systems, man, and cybernetics* (6), 610-621.

Harwood, D., Ojala, T., Pietikäinen, M., Kelman, S. & Davis, L. (1995). "Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions", *Pattern Recognition Letters*, 16(1), 1-10.

Hasoon, J. N. & Hassan, R. (2019). "Face Image Retrieval Based on Fireworks Algorithm", in *2019 1st AL-Noor International Conference for Science and Technology (NICST)*. IEEE, 94-99.

Hassan, R. Q., Sultani, Z. N. & Dhannoon, B. N. (2023). "Content-Based Image Retrieval System using Color Moment and Bag of Visual Words with Local Binary Pattern", *Karbala International Journal of Modern Science*, 9(1), 7.

Hayman, E., Caputo, B., Fritz, M. & Eklundh, J.-O. (2004). "On the significance of real-world conditions for material classification", in *European conference on computer vision*. Springer, 253-266.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770-778.

He, S., Ruan, J., Long, Y., Wang, J., Wu, C., Ye, G., Zhou, J., Yue, J. & Zhang, Y. (2018). "Combining deep learning with traditional features for classification and segmentation of pathological images of breast cancer", in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*. IEEE, 1, 3-6.

Heikkilä, M., Pietikäinen, M. & Schmid, C. (2009). "Description of interest regions with local binary patterns", *Pattern recognition*, 42(3), 425-436.

Holland, J. (1975). "Adaptation in neural and artificial systems", *Ann Arbor: University of Michigan Press*.

Hsu, C.-Y., Lu, C.-S. & Pei, S.-C. (2012). "Image feature extraction in encrypted domain with privacy-preserving SIFT", *IEEE transactions on image processing*, 21(11), 4593-4607.

Huang, D., Shan, C., Ardabilian, M., Wang, Y. & Chen, L. (2011). "Local binary patterns and its application to facial image analysis: a survey", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6), 765-781.

Huang, H. K. (2011). "*PACS and imaging informatics: basic principles and applications*", John Wiley & Sons.

Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J. & Zabih, R. (1997). "Image indexing using color correlograms", in *Proceedings of IEEE computer society conference on Computer Vision and Pattern Recognition*. IEEE, 762-768.

Huiskes, M. J., Thomee, B. & Lew, M. S. (2010). "New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative", in *Proceedings of the international conference on Multimedia information retrieval*. 527-536.

Hull, D. (1993). "Using statistical testing in the evaluation of retrieval experiments", in *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. 329-338.

Humeau-Heurtier, A. (2019). "Texture feature extraction methods: A survey", *IEEE access*, 7, 8975-9000.

Imdad, F., Haq, S. U., Rasool, I., Wasimuddin, M. & Farooq, M. O. (2024). "Content-Based Image Retrieval Established on Deep and Handcrafted Features", *The Sciencetech*, 5(2), 1-12.

ISO/MPEG-7 (2001) *2001: Information technology—Multimedia content description interface—Part 3: Visual.*

Issaoui, I., Alohali, M. A., Mtouaa, W., Alotaibi, F. A., Mahmud, A. & Assiri, M. (2024). "Archimedes Optimization Algorithm With Deep Learning Assisted Content-Based Image Retrieval in Healthcare Sector", *IEEE Access*, 12, 29768-29777.

Jafarinejad, F. & Farzbood, R. (2021). "Relevance Feedback-based Image Retrieval using Particle Swarm Optimization", *Journal of AI and Data Mining*, 9(2), 245-257.

Jain, A. & Healey, G. (1998). "A multiscale representation including opponent color features for texture recognition", *IEEE Transactions on Image Processing*, 7(1), 124-128.

Jain, K. & Bhadauria, S. S. (2016). "Enhanced content based image retrieval using feature selection using teacher learning based optimization", *Int J Comput Sci Inf Secur (IJCSIS)*, 14, 1052-1057.

Jan, M. M., Zainal, N. & Jamaludin, S. (2020). "Region of interest-based image retrieval techniques: a review", *IAES International Journal of Artificial Intelligence*, 9(3), 520.

Janarthanam, S. & Sukumaran, S. (2016). "Semi supervised soft label propagation algorithm for CBIR", in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 1-5.

Janati Idrissi, B., El Ogri, O. & EL-Mekkaoui, J. (2024). "A new retrieval system based on quaternion radial orthogonal Jacobi moments for biomedical color images", *Multimedia Tools and Applications*, 1-25.

Jardim, S., António, J., Mora, C. & Almeida, A. (2022). "A novel trademark image retrieval system based on multi-feature extraction and deep networks", *Journal of imaging*, 8(9), 238.

Jhaveri, R. H., Revathi, A., Ramana, K., Raut, R. & Dhanaraj, R. K. (2022). "A review on machine learning strategies for real-world engineering applications", *Mobile Information Systems*, 2022(1), 1833507.

Jiang, D. (2021). "Image Feature Fusion and Fisher Coding based Method for CBIR", in *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*. IEEE, 503-508.

Jin, H., Liu, Q., Lu, H. & Tong, X. (2004). "Face detection using improved LBP under Bayesian framework", in *Third International Conference on Image and Graphics (ICIG'04)*. IEEE, 306-309.

Johnson, S. C. (1967). "Hierarchical clustering schemes", *Psychometrika*, 32(3), 241-254.

Jourdan, L., Basseur, M. & Talbi, E.-G. (2009). "Hybridizing exact methods and metaheuristics: A taxonomy", *European Journal of Operational Research*, 199(3), 620-629.

Kale, M. & Mukhopadhyay, S. (2022). "Efficient color image retrieval method using deep stacked sparse autoencoder", *Journal of Electronic Imaging*, 31(2), 023003-023003.

Kamali, H. R., Sadegheih, A., Vahdat-Zad, M. A. & Khademi-Zare, H. (2015). "Immigrant population search algorithm for solving constrained optimization problems", *Applied Artificial Intelligence*, 29(3), 243-258.

Kapadia, M. R. & Paunwala, C. N. (2021). "Multi-Channel Convolution Neural Network for Accurate CBMIR System with Reduced Semantic Gap", *International Journal of Advanced Research in Engineering and Technology*, 12(1).

Karakasis, E. G., Amanatiadis, A., Gasteratos, A. & Chatzichristofis, S. A. (2015). "Image moment invariants as local features for content based image retrieval using the bag-of-visual-words model", *Pattern Recognition Letters*, 55, 22-27.

Karanwal, S. (2024). "Robust And Discriminant Local Color Pattern (RADLCP): A novel color descriptor for face recognition", *International Journal of Hybrid Intelligent Systems*, 20(1), 23-39.

Karanwal, S. & Diwakar, M. (2023). "Triangle and orthogonal local binary pattern for face recognition", *Multimedia Tools and Applications*, 82(23), 36179-36205.

Kashif, M., Raja, G. & Shaukat, F. (2020). "An efficient content-based image retrieval system for the diagnosis of lung diseases", *Journal of digital imaging*, 33(4), 971-987.

Kather, J. N., Weis, C.-A., Bianconi, F., Melchers, S. M., Schad, L. R., Gaiser, T., Marx, A. & Zöllner, F. G. (2016). "Multi-class texture analysis in colorectal cancer histology", *Scientific reports*, 6(1), 1-11.

Kato, T. (1992). "Database architecture for content-based image retrieval", in *image storage and retrieval systems*. International Society for Optics and Photonics, 1662, 112-123.

Ke-Lin Du, M. N. S. S. (2016). "*Search and optimization by metaheuristics: Techniques and Algorithms Inspired by Nature*". Switzerland Birkhäuser Cham.

Ke, P., Cai, M., Wang, H. & Chen, J. (2018). "A novel face recognition algorithm based on the combination of LBP and CNN", in *2018 14th IEEE International Conference on Signal Processing (ICSP)*. IEEE, 539-543.

Kelishadrokhi, M. K., Ghattaei, M. & Fekri-Ershad, S. (2023). "Innovative local texture descriptor in joint of human-based color features for content-based image retrieval", *Signal, Image and Video Processing*, 17(8), 4009-4017.

Kelly, P. M. & Cannon, T. M. (1994). "CANDID: Comparison algorithm for navigating digital image databases", in *Seventh International Working Conference on Scientific and Statistical Database Management*. IEEE, 252-258.

Kennedy, J. & Eberhart, R. (1995). "Particle swarm optimization", in *Proceedings of ICNN'95-international conference on neural networks*. IEEE, 4, 1942-1948.

Keysers, D., Dahmen, J. r., Ney, H., Wein, B. B. & Lehmann, T. M. (2003). "Statistical framework for model-based image retrieval in medical applications", *Journal of Electronic Imaging*, 12(1), 59-68.

Khadilkar, S. P. (2022). "Colon cancer detection using hybrid features and genetically optimized neural network classifier", *International Journal of Image and Graphics*, 22(02), 2250024.

Khan, A., Rajvee, M. H., Deekshatulu, B. & Pratap Reddy, L. (2023). "A fused LBP texture descriptor-based image retrieval system", *Advances in Signal Processing, Embedded Systems and IoT: Proceedings of Seventh ICMEET-2022*Springer, 145-154.

Khediri, N., Ammar, M. B. & Kherallah, M. (2021). "Comparison of image segmentation using different color spaces", in *2021 IEEE 21st International Conference on Communication Technology (ICCT)*. IEEE, 1188-1192.

Khokher, A. & Talwar, R. (2012). "Content-based image retrieval: Feature extraction techniques and applications", in *International conference on recent advances and future trends in information technology (iRAFIT2012)*. 9-14.

Kim, M., Yun, J., Cho, Y., Shin, K., Jang, R., Bae, H.-j. & Kim, N. (2019). "Deep learning in medical imaging", *Neurospine*, 16(4), 657.

Kirkpatrick, S., Gelatt Jr, C. D. & Vecchi, M. P. (1983). "Optimization by simulated annealing", *science*, 220(4598), 671-680.

Koza, J. R. (1992). "*Genetic programming: the movie*". Cambridge, MA: MIT Press.

Krishnan, A., Rajesh, S. & Shylaja, S. (2021). "Text-based image retrieval using captioning", in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, 1-5.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks", *Advances in neural information processing systems*, 25.

Kumar, A., Kim, J., Cai, W., Fulham, M. & Feng, D. (2013). "Content-based medical image retrieval: a survey of applications to multidimensional and multimodality data", *Journal of digital imaging*, 26, 1025-1039.

Kumar, A. R. & Saravanan, D. (2013). "Content based image retrieval using color histogram", *International journal of computer science and information technologies*, 4(2), 242-245.

Kumar, G. & Bhatia, P. K. (2014). "A detailed review of feature extraction in image processing systems", in *2014 Fourth international conference on advanced computing & communication technologies*. IEEE, 5-12.

Kumar, M. D., Babaie, M., Zhu, S., Kalra, S. & Tizhoosh, H. R. (2017). "A comparative study of CNN, BoVW and LBP for classification of histopathological images", in *2017 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 1-7.

Kumar, N. R. & Kumar, R. (2023). "Efficient medical image retrieval system using Geometric Invariant Point Bilateral Transformation (GIPBT)", *Measurement: Sensors*, 27, 100705.

Kumar, R. & Murthy, N. (2025). "Enhanced Content Based Image Retrieval Using Integrated Color and Texture Features", *IJSAT-International Journal on Science and Technology*, 16(1), 1-13.

Kumar, Y. R., Narayanappa, C. & Dayananda, P. (2020). "Weighted full binary tree-sliced binary pattern: An RGB-D image descriptor", *Heliyon*, 6(5), 1-18.

Kumari, V. V. (2015). "Improved Hill Climbing Based Segmentation (IHCBS) technique for CBIR system", in *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, 1-10.

Kunttu, I., Lepisto, L., Rauhamaa, J. & Visa, A. (2004). "Multiscale Fourier descriptor for shape-based image retrieval", in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*: IEEE, 2, 765-768.

Kushwaha, P. & Welekar, R. R. (2016). "Feature Selection for Image Retrieval based on Genetic Algorithm", *Int. J. Interact. Multim. Artif. Intell.*, 4(2), 16-21.

Kwitt, R. (2021) *Salzburg Texture Image Dataset,* . [Created 5 November 2021]

Lahdenoja, O., Laiho, M. & Paasio, A. (2005). "Reducing the feature vector length in local binary pattern based face recognition", in *IEEE International Conference on Image Processing 2005*. IEEE, 2, II-914.

Lakmann, R. (1998). "Barktex benchmark database of color textured images", *Koblenz-Landau University*.

Lam, M. O., Disney, T., Raicu, D. S., Furst, J. & Channin, D. S. (2007). "BRISC—an open source pulmonary nodule image retrieval framework", *Journal of digital imaging*, 20, 63-71.

Lan, R. & Zhou, Y. (2016). "Medical image retrieval via histogram of compressed scattering coefficients", *IEEE journal of biomedical and health informatics*, 21(5), 1338-1346.

Latif, A., Rasheed, A., Sajid, U., Ahmed, J., Ali, N., Ratyal, N. I., Zafar, B., Dar, S. H., Sajid, M. & Khalil, T. (2019). "Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review", *Mathematical problems in engineering*, 2019(1), 9658350.

Lew, M. S., Sebe, N., Djeraba, C. & Jain, R. (2006). "Content-based multimedia information retrieval: State of the art and challenges", *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2(1), 1-19.

Li, H., Wu, G. & Zheng, W.-S. (2021a). "Combined depth space based architecture search for person re-identification", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6729-6738.

Li, J. & Wang, J. Z. (2003). "Automatic linguistic indexing of pictures by a statistical modeling approach", *IEEE Transactions on pattern analysis and machine intelligence*, 25(9), 1075-1088.

Li, W., Duan, L., Xu, D. & Tsang, I. W.-H. (2011). "Text-based image retrieval using progressive multi-instance learning", in *2011 international conference on computer vision*. IEEE, 2049-2055.

Li, X., Yang, J. & Ma, J. (2021b). "Recent developments of content-based image retrieval (CBIR)", *Neurocomputing*, 452, 675-689.

Li, Y., Zhang, J., Chen, M., Lei, H., Luo, G. & Huang, Y. (2019). "Shape based local affine invariant texture characteristics for fabric image retrieval", *Multimedia Tools and Applications*, 78, 15433-15453.

Li, Z., Liu, F., Yang, W., Peng, S. & Zhou, J. (2021c). "A survey of convolutional neural networks: analysis, applications, and prospects", *IEEE transactions on neural networks and learning systems*, 33(12), 6999-7019.

Liang, T., Jin, Y., Liu, W., Wang, T., Feng, S. & Li, Y. (2024). "Bridging the Gap: Multi-level Cross-modality Joint Alignment for Visible-infrared Person Re-identification", *IEEE Transactions on Circuits and Systems for Video Technology*, 34(8), 7683-7698.

Liang, X., Yang, J., Yi, T., Yang, K., Zhang, W., Fan, Y. & Li, P. (2020). "Optimized Algorithm Of Human 3D Motion Recognition Based On HOG And LBP", in *2020 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*. IEEE, 66-71.

Liao, S., Hu, Y., Zhu, X. & Li, S. Z. (2015). "Person re-identification by local maximal occurrence representation and metric learning", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2197-2206.

Liao, S., Law, M. W. & Chung, A. C. (2009). "Dominant local binary patterns for texture classification", *IEEE transactions on image processing*, 18(5), 1107-1118.

Liao, S., Zhu, X., Lei, Z., Zhang, L. & Li, S. Z. (2007). "Learning multi-scale block local binary patterns for face recognition", in *Advances in Biometrics: International Conference, ICB 2007, Seoul, Korea, August 27-29, 2007. Proceedings*. Springer, 828-837.

Liu, J., Bi, C., Chen, H., Heidari, A. A. & Chen, H. (2025). "Triangular-based sine cosine algorithm for global search and feature selection", *Scientific Reports*, 15(1), 12992.

Liu, L., Lao, S., Fieguth, P. W., Guo, Y., Wang, X. & Pietikäinen, M. (2016). "Median robust extended local binary pattern for texture classification", *IEEE Transactions on Image Processing*, 25(3), 1368-1381.

Liu, L., Zhao, L., Long, Y., Kuang, G. & Fieguth, P. (2012). "Extended local binary patterns for texture classification", *Image and Vision Computing*, 30(2), 86-99.

Liu, P., Guo, J.-M., Chamnongthai, K. & Prasetyo, H. (2017a). "Fusion of color histogram and LBP-based features for texture image retrieval and classification", *Information Sciences*, 390, 95-111.

Liu, P., Guo, J.-M., Wu, C.-Y. & Cai, D. (2017b). "Fusion of deep learning and compressed domain features for content-based image retrieval", *IEEE Transactions on Image Processing*, 26(12), 5706-5717.

Liu, Y., Zhang, D. & Lu, G. (2008). "Region-based image retrieval with high-level semantics using decision tree learning", *Pattern Recognition*, 41(8), 2554-2570.

Liu, Y., Zhang, D., Lu, G. & Ma, W.-Y. (2007). "A survey of content-based image retrieval with high-level semantics", *Pattern recognition*, 40(1), 262-282.

Lourenço, H. R., Martin, O. C. & Stützle, T. (2003). "Iterated local search", *Handbook of metaheuristics*Springer, 320-353.

Lu, W., Swaminathan, A., Varna, A. L. & Wu, M. (2009). "Enabling search over encrypted multimedia databases", in *Media Forensics and Security*. SPIE, 7254, 404-414.

Luft, J. & Schiewe, J. (2021). "Content-based Image Retrieval for Map Georeferencing", in *Proceedings of the ICA*. Copernicus GmbH, 4, 1-8.

Ma, W.-Y. & Manjunath, B. S. (1999). "Netra: A toolbox for navigating large image databases", *Multimedia systems*, 7, 184-198.

Mäenpää, T. (2003). "*The local binary pattern approach to texture analysis—extensions and applications*" Ph.D thesis. University of Oulu.

Mäenpää, T., Ojala, T., Pietikäinen, M. & Soriano, M. (2000). "Robust texture classification by subsets of local binary patterns", in *Proceedings of the 15th International Conference on Pattern Recognition*. Barcelona, Spain, 947–950.

Mäenpää, T. & Pietikäinen, M. (2004). "Classification with color and texture: jointly or separately?", *Pattern recognition*, 37(8), 1629-1640.

Mahmood, A., Imran, M., Irtaza, A., Abbas, Q., Dhahri, H., Othman, E. M. A., Malik, A. J. & Abbasi, A. A. (2022). "Hybrid evolutionary algorithm based relevance feedback approach for image retrieval", *CMC-Comput Mater Contin*, 70(1), 963-979.

Mane, P. & Bawane, N. (2014). "An approach to explore the role of color models and color descriptors in the optimization of semantic gap in content based image retrieval", *International Journal of Computer Applications*, 104(14), 9-16.

Manning, C. D. (2008). "Introduction to information retrieval". Cambridge university press.

Marcus, D. S., Fotenos, A. F., Csernansky, J. G., Morris, J. C. & Buckner, R. L. (2010). "Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults", *Journal of cognitive neuroscience*, 22(12), 2677-2684.

Marcus, D. S., Wang, T. H., Parker, J., Csernansky, J. G., Morris, J. C. & Buckner, R. L. (2007). "Open Access Series of Imaging Studies (OASIS): cross-sectional MRI data in

young, middle aged, nondemented, and demented older adults", *Journal of cognitive neuroscience*, 19(9), 1498-1507.

Mardi, K. S., Mawardi, V. C. & Perdana, N. J. (2019). "KNN classification for CBIR with color moments, connected regions, discrete wavelet transform", in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*. IEEE, 6, 238-243.

Martí, R. & Reinelt, G. (2011). "*The linear ordering problem: exact and heuristic methods in combinatorial optimization*", *175*, Springer Science & Business Media.

Materka, A. & Strzelecki, M. (1998). "Texture analysis methods–a review", *Technical university of lodz, institute of electronics, COST B11 report, Brussels*, 10(1.97), 4968.

Mathews, A., Sejal, N. & Venugopal, K. (2022). "Analysis of content based image retrieval using deep feature extraction and similarity matching", *International Journal of Advanced Computer Science and Applications*, 13(12), 646-655.

Maturana, D., Mery, D. & Soto, A. (2011). "Learning discriminative local binary patterns for face recognition", in *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. IEEE, 470-475.

Mehta, R. & Egiazarian, K. (2016). "Dominant rotated local binary patterns (DRLBP) for texture classification", *Pattern Recognition Letters*, 71, 16-22.

Mezaris, V., Kompatsiaris, I. & Strintzis, M. G. (2004). "Region-based image retrieval using an object ontology and relevance feedback", *EURASIP Journal on Advances in Signal Processing*, 2004, 1-16.

Mi, Z. & Yang, B. "Heuristic algorithms", *Cornell University Computational Optimization Open Textbook*.

Mihoubi, S., Losson, O., Mathon, B. & Macaire, L. (2018). "Spatio-spectral binary patterns based on multispectral filter arrays for texture classification", *Journal of the Optical Society of America A*, 35(9), 1532-1542.

Min, H. & Shuangyuan, Y. (2010). "Overview of content-based image retrieval with high-level semantics", in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. IEEE, 6, V6-312-V316-316.

Mirjalili, S. (2016). "SCA: a sine cosine algorithm for solving optimization problems", *Knowledge-based systems*, 96, 120-133.

Mori, Y., Takahashi, H. & Oka, R. (1999). "Image-to-word transformation based on dividing and vector quantizing images with words", in *First international workshop on multimedia intelligent storage and retrieval management*. Citeseer, 2, 1–9.

Mosbah, M. & Boucheham, B. (2014). "Relevance feedback within CBIR systems", *International Journal of Computer, Information Science and Engineering*, 8(4), 568-572.

Mosbah, M. & Boucheham, B. (2017). "Distance selection based on relevance feedback in the context of CBIR using the SFS meta-heuristic with one round", *Egyptian Informatics Journal*, 18(1), 1-9.

Mujawar, S., Ladkhan, N. H., Infotech, K. & Sambrekar, K. (2014). "A Content Based Image Retrieval System for diagnosing Agricultural Plant Diseases", *International Journal of Engineering Research*, 3(3), 878-880.

Murthy, V., Vamsidhar, E., Kumar, J. S. & Rao, P. S. (2010a). "Content based image retrieval using Hierarchical and K-means clustering techniques", *International Journal of Engineering Science and Technology*, 2(3), 209-212.

Murthy, V., Vamsidhar, E., Rao, P. S., Samuel, G. & Raju, V. (2010b). "Application of hierarchical and K-means techniques in content based image retrieval", *International Journal of Engineering Science and Technology*, 2(5), 749-755.

Muslihah, I., Muqorobin, M., Rokhmah, S., Rais, R. & Akbar, N. (2020). "Texture characteristic of local binary pattern on face recognition with probabilistic linear discriminant analysis", *International Journal of Computer and Information System*, 1(1), 341472.

Naik, A. & Satapathy, S. C. (2021). "A comparative study of social group optimization with a few recent optimization algorithms", *Complex & Intelligent Systems*, 7(1), 249-295.

Nalini, P. & Malleswari, B. (2016). "Review on content based image retrieval: From its origin to the new age", *Int. J. Res. Stud. Sci. Eng. Technol*, 3(2), 18-41.

Nanni, L., Lumini, A. & Brahnam, S. (2012). "Survey on LBP based texture descriptors for image classification", *Expert Systems with Applications*, 39(3), 3634-3641.

Naresh, Y. & Nagendraswamy, H. (2016). "Classification of medicinal plants: an approach using modified LBP with symbolic representation", *Neurocomputing*, 173, 1789-1797.

Narwade, J. & Kumar, B. (2016). "Local and global color histogram feature for color content-based image retrieval system", in *Proceedings of the International Congress on Information and Communication Technology: ICICT 2015, Volume 1*. Springer, 293-300.

Nava, R., Cristóbal, G. & Escalante-Ramírez, B. (2012). "A comprehensive study of texture analysis based on local binary patterns", in *Optics, Photonics, and Digital Technologies for Multimedia Applications II*. SPIE, 8436, 125-136.

Ndung'u, S., Grobler, T., Wijnholds, S. J. & Azzopardi, G. (2025). "Content-based image retrieval using COSFIRE descriptors with application to radio astronomy", *Monthly Notices of the Royal Astronomical Society*, 537(4), 3286-3297.

Nemade, S. & Sonavane, S. (2020). "Automatic feature extraction for CBIR and image annotation applications", in *Computing in Engineering and Technology: Proceedings of ICCET 2019*. Springer, 557-566.

Nene, S. A., Nayar, S. K. & Murase, H. (1996). *"Columbia object image library (coil-20)"*CUCS-006-96, T. r.

Nesmachnow, S. (2014). "An overview of metaheuristics: accurate and efficient methods for optimisation", *International Journal of Metaheuristics*, 3(4), 320-347.

Nguyen, H. V., Bai, L. & Shen, L. (2009). "Local gabor binary pattern whitened pca: A novel approach for face recognition from single image per person", in *Advances in Biometrics: Third International Conference, ICB 2009, Alghero, Italy, June 2-5, 2009. Proceedings 3*. Springer, 269-278.

Nhi, N. T. U., Thi, N. T. U. & Han, P. T. A. (2020). "A SEMANTIC-BASED IMAGE RETRIEVAL SYSTEM USING A HYBRID METHOD K-MEANS AND K-NEAREST-NEIGHBOR", in *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae. Sectio Computatorica*. 51, 253–274.

Niño-Adan, I., Manjarres, D., Landa-Torres, I. & Portillo, E. (2021). "Feature weighting methods: A review", *Expert Systems with Applications*, 184, 115424.

Oh, J. H., Yang, Y. & El Naqa, I. (2010). "Adaptive learning for relevance feedback: application to digital mammography", *Medical physics*, 37(8), 4432-4444.

Ohashi, T., Aghbari, Z. & Makinouchi, A. (2003). "Hill-climbing algorithm for efficient color-based image segmentation", in *IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*. 17-22.

Ojala, T., Pietikäinen, M. & Harwood, D. (1996). "A comparative study of texture measures with classification based on featured distributions", *Pattern recognition*, 29(1), 51-59.

Ojala, T., Pietikainen, M. & Maenpaa, T. (2002). "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", *IEEE Transactions on pattern analysis and machine intelligence*, 24(7), 971-987.

Okamoto, T. & Hirata, H. (2013). "Global optimization using a multipoint type quasi-chaotic optimization method", *Applied Soft Computing*, 13(2), 1247-1264.

Ouni, A., Chateau, T., Royer, E., Chevaldonné, M. & Dhome, M. (2022). "A new cbir model using semantic segmentation and fast spatial binary encoding", in *International Conference on Computational Collective Intelligence*. Springer, 437-449.

Park, D. & Hwang, Y. (2024). "Efficient Image Retrieval Using Hierarchical K-Means Clustering", *Sensors*, 24(8), 2401.

Park, J.-S. & Kim, T. (2005). "Shape-based image retrieval using invariant features", in *Advances in Multimedia Information Processing-PCM 2004: 5th Pacific Rim Conference on Multimedia, Tokyo, Japan, November 30-December 3, 2004. Proceedings, Part II 5*. Springer, 146-153.

Park, J., An, Y., Jeong, I., Kang, G. & Pankoo, K. (2007). "Image indexing using spatial multi-resolution color correlogram", in *2007 IEEE International Workshop on Imaging Systems and Techniques*. IEEE, 1-4.

Pathak, D. & Raju, U. (2021). "Content-based image retrieval using feature-fusion of GroupNormalized-Inception-Darknet-53 features and handcraft features", *Optik*, 246, 167754.

Patil, P. B. & Kokare, M. B. (2013). "Texture image retrieval using greedy method", in *Proceedings of International Conference on Advances in Computing*. Springer, 885-891.

Pentland, A., Picard, R. W. & Sclaroff, S. (1996). "Photobook: Content-based manipulation of image databases", *International journal of computer vision*, 18(3), 233-254.

Petscharnig, S., Lux, M. & Chatzichristofis, S. (2017). "Dimensionality reduction for image features using deep learning and autoencoders", in *Proceedings of the 15th international workshop on content-based multimedia indexing*. 1-6.

Pietikäinen, M., Hadid, A., Zhao, G. & Ahonen, T. (2011). "*Computer vision using local binary patterns*", *40*. London, U.K: Springer Science & Business Media.

Pietikäinen, M. & Zhao, G. (2015). "Two decades of local binary patterns: A survey", *Advances in independent component analysis and learning machines*Elsevier, 175-210.

Po, L.-M. & Wong, K.-M. (2004). "A new palette histogram similarity measure for MPEG-7 dominant color descriptor", in *2004 International Conference on Image Processing, 2004. ICIP'04.*: IEEE, 3, 1533-1536.

Porebski, A., Hoang, V. T., Vandenbroucke, N. & Hamad, D. (2018). "Multi-color space local binary pattern-based feature selection for texture classification", *Journal of Electronic Imaging*, 27(1), 011010-011010.

Porebski, A., Vandenbroucke, N., Macaire, L. & Hamad, D. (2014). "A new benchmark image test suite for evaluating colour texture classification schemes", *Multimedia Tools and Applications*, 70(1), 543-556.

Pradhan, J., Ajad, A., Pal, A. K. & Banka, H. (2019). "Multi-level colored directional motif histograms for content-based image retrieval", *The Visual Computer*, 36(9), 1847-1868.

Pradhan, J., Kumar, S., Pal, A. K. & Banka, H. (2018). "Texture and color visual features based CBIR using 2D DT-CWT and histograms", in *Mathematics and Computing: 4th International Conference, ICMC 2018, Varanasi, India, January 9-11, 2018, Revised Selected Papers 4*. Springer, 84-96.

Putzu, L., Piras, L. & Giacinto, G. (2020). "Convolutional neural networks for relevance feedback in content based image retrieval: A content based image retrieval system that exploits convolutional neural networks both for feature extraction and for relevance feedback", *Multimedia Tools and Applications*, 79(37), 26995-27021.

Qin, A. K., Huang, V. L. & Suganthan, P. N. (2008). "Differential evolution algorithm with strategy adaptation for global numerical optimization", *IEEE transactions on Evolutionary Computation*, 13(2), 398-417.

Raghuwanshi, G., Mishra, N. & Sharma, S. (2012). "Content based image retrieval using implicit and explicit feedback with interactive genetic algorithm", *International Journal of Computer Applications*, 43(16), 8-14.

Raja, D. & Karthikeyan, M. (2022). "Content based image retrieval using reptile search algorithm with deep learning for agricultural crops", in *2022 7th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 1038-1043.

Ramesh Babu, P. & Reddy, E. S. (2018). "A comprehensive survey on semantic based image retrieval systems for cyber forensics", *International Journal of Computer Sciences and Engineering*, 6(8), 245-250.

Ramola, A., Shakya, A. K. & Van Pham, D. (2020). "Study of statistical methods for texture analysis and their modern evolutions", *Engineering Reports*, 2(4), e12149.

Rao, M. B., Rao, B. P. & Govardhan, A. (2011a). "CTDCIRS: content based image retrieval system based on dominant color and texture features", *International Journal of Computer Applications*, 18(6), 40-46.

Rao, R. V., Savsani, V. J. & Vakharia, D. P. (2011b). "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems", *Computer-aided design*, 43(3), 303-315.

Rashedi, E., Nezamabadi-Pour, H. & Saryazdi, S. (2009). "GSA: a gravitational search algorithm", *Information sciences*, 179(13), 2232-2248.

Rechenberg, I. (1973). "Evolutionsstrategie", *Optimierung technischer Systeme nach Prinzipien derbiologischen Evolution*.

Reddi, K. K. & Enireddy, V. (2016). "Cuckoo search framework for feature selection and classifier optimization in compressed medical image retrieval", *i-manager's Journal on Image Processing*, 3(1), 1.

Rizk-Allah, R. M. & Hassanien, A. E. (2023). "A comprehensive survey on the sine–cosine optimization algorithm", *Artificial Intelligence Review*, 56(6), 4801-4858.

Rosenfeld, A. (1975). *"Visual texture analysis: An overview*".

Roshdi, A. & Roohparvar, A. (2015). "Information retrieval techniques and applications", *International Journal of Computer Networks and Communications Security*, 3(9), 373-377.

Rout, N. K., Atulkar, M. & Ahirwal, M. K. (2021). "A review on content-based image retrieval system: present trends and future challenges", *International Journal of Computational Vision and Robotics*, 11(5), 461-485.

Rui, Y., Huang, T. S. & Mehrotra, S. (1997). "Content-based image retrieval with relevance feedback in MARS", in *Proceedings of international conference on image processing*. IEEE, 2, 815-818.

S. A. Nene, S. K. N., and H. Murase (1996). "Colombia object image library (COIL-100)", *Dept.Of Computer Science, Colombia University, Technical report CUCS-006-96*, 7, 1-6.

Salmi, M. & Boucheham, B. (2014). "Content based image retrieval based on cell color coherence vector (Cell-CCV)", in *2014 4th International Symposium ISKO-Maghreb: Concepts and Tools for knowledge Management (ISKO-Maghreb)*. IEEE, 1-5.

Sameer, F. O., Al-obaidi, M. J., Al-bassam, W. W. & Ad'hiah, A. H. (2021). "Multi-objectives TLBO hybrid method to select the related risk features with rheumatism disease", *Neural Computing and Applications*, 33, 9025-9034.

Samet, H. (1984). "The quadtree and related hierarchical data structures", *ACM Computing Surveys (CSUR)*, 16(2), 187-260.

Saminathan, K., Amsavalli, S. & Devi, M. C. (2024). "Content based medical image retrieval using multi-feature extraction and patch Sorensen similarity indexing technique", *International Journal of Bioinformatics Research and Applications*, 20(5), 439-467.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510-4520.

Saravanan, A. & Sathiamoorthy, S. (2019). "Autocorrelation based Chordiogram image descriptor for image retrieval", in *2019 International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 1990-1996.

Sastry, C. S., Ravindranath, M., Pujari, A. K. & Deekshatulu, B. L. (2007). "A modified Gabor function for content based image retrieval", *Pattern Recognition Letters*, 28(2), 293-300.

Schuld, M., Sinayskiy, I. & Petruccione, F. (2015). "An introduction to quantum machine learning", *Contemporary Physics*, 56(2), 172-185.

Shabir, A., Ahmed, K. T., Mahmood, A., Garay, H., Prado González, L. E. & Ashraf, I. (2025). "Deep image features sensing with multilevel fusion for complex convolution neural networks & cross domain benchmarks", *PloS one*, 20(3), e0317863.

Shahamat, H. & Pouyan, A. A. (2015). "Feature selection using genetic algorithm for classification of schizophrenia using fMRI data", *Journal of AI and Data Mining*, 3(1), 30-37.

Shaik, K. B., Ganesan, P., Kalist, V., Sathish, B. & Jenitha, J. M. M. (2015). "Comparative study of skin color detection and segmentation in HSV and YCbCr color space", *Procedia Computer Science*, 57, 41-48.

Shan, C. (2012). "Learning local binary patterns for gender classification on real-world face images", *Pattern recognition letters*, 33(4), 431-437.

Shatnawi, M. Q., Alrousan, M. & Amareen, S. (2020). "A new approach for content-based image retrieval for medical applications using low-level image descriptors", *International Journal of Electrical and Computer Engineering*, 10(4), 4363.

Shih, J.-L. & Chen, L.-H. (2002). "Color image retrieval based on primitives of color moments", in *VISUAL*. Springer, 88-94.

Shoib, A. M., Summaira, J., Wang, C. & Jabbar, A. (2023). "Methods and advancement of content-based fashion image retrieval: A Review", *arXiv preprint arXiv:2303.17371*.

Shukla, A. K. & Kanungo, S. (2020). "An automated face retrieval system using grasshopper optimization algorithm-based feature selection method", in *Emerging Trends in Computing and Expert Technology*. Springer, 492-502.

Shyu, C.-R., Brodley, C. E., Kak, A. C., Kosaka, A., Aisen, A. M. & Broderick, L. S. (1999). "ASSERT: A physician-in-the-loop content-based retrieval system for HRCT image databases", *Computer vision and image understanding*, 75(1-2), 111-132.

Simon, D. (2008). "Biogeography-based optimization", *IEEE transactions on evolutionary computation*, 12(6), 702-713.

Simonyan, K. & Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*.

Singh, B. & Ahmad, W. (2014). "Content based image retrieval: a review paper", *International Journal of Computer Science and Mobile Computing*, 3(5), 769-775.

Singh, G. & Yogi, K. K. (2023). "A Detailed Review on Recognition of Plant Disease Using Intelligent Image Retrieval Techniques", *IJCSNS*, 23(9), 77.

Singh, R. K., Patidar, K., Kushwah, R. & Chouhan, S. (2017). "Content image retrieval using combination of dominant and LBP based features", *ACCENTS Transactions on Image Processing and Computer Vision*, 3(8), 13.

Singh, S. M. & Hemachandran, K. (2012). "Image retrieval based on the combination of color histogram and color moment", *International journal of computer applications*, 58(3).

Singh, S. R., Dubey, S. R., MS, S., Ventrapragada, S. & Dasharatha, S. S. (2024). "Joint triplet autoencoder for histopathological colon cancer nuclei retrieval", *Multimedia Tools and Applications*, 83(1), 1063-1082.

Singhania, U. & Tripathy, B. (2021). "Text-based image retrieval using deep learning", *Encyclopedia of Information Science and Technology, Fifth Edition*IGI Global, 87-97.

Sklansky, J. (1978). "Image segmentation and feature extraction", *IEEE Transactions on Systems, Man, and Cybernetics*, 8(4), 237-247.

Smeulders, A. W., Worring, M., Santini, S., Gupta, A. & Jain, R. (2000). "Content-based image retrieval at the end of the early years", *IEEE Transactions on pattern analysis and machine intelligence*, 22(12), 1349-1380.

Smith, J. R. & Chang, S.-F. (1997). "VisualSEEk: a fully automated content-based image query system", in *Proceedings of the fourth ACM international conference on Multimedia*. 87-98.

Somnugpong, S. & Khiewwan, K. (2016). "Content-based image retrieval using a combination of color correlograms and edge direction histogram", in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 1-5.

Song, X., Chen, Y., Feng, Z.-H., Hu, G., Zhang, T. & Wu, X.-J. (2019). "Collaborative representation based face classification exploiting block weighted LBP and analysis dictionary learning", *Pattern Recognition*, 88, 127-138.

Soni, P., Lamba, V. K. & Kumar, S. (2021). "Optimized Color Correlogram based CBIR to beat the hole of Information and Interpretation", in *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE, 387-392.

Sotoodeh, M., Moosavi, M. R. & Boostani, R. (2019). "A novel adaptive LBP-based descriptor for color image retrieval", *Expert Systems with Applications*, 127, 342-352.

Souid, A., Othman, S. B., Hamroun, M. & Sakli, H. (2023). "Full Interpretability CBMIR to Help Minimize Radiologist Analysis Search Time", in *2023 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 1136-1141.

Srivastava, D., Rajitha, B., Agarwal, S. & Singh, S. (2020). "Pattern-based image retrieval using GLCM", *Neural computing and applications*, 32, 10819-10832.

Srivastava, D., Singh, S. S., Rajitha, B., Verma, M., Kaur, M. & Lee, H.-N. (2023). "Content-based Image Retrieval: A Survey on Local and Global Features Selection, Extraction, Representation and Evaluation Parameters", *IEEE Access*.

Storn, R. M. & Price, K. (1995). "Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces", *Journal of Global Optimization* 23(1).

Stricker, M. A. & Orengo, M. (1995). "Similarity of color images", in *Storage and retrieval for image and video databases III*. SPiE, 2420, 381-392.

Subramanian, M., Lingamuthu, V., Venkatesan, C. & Perumal, S. (2022). "Content-Based Image Retrieval Using Colour, Gray, Advanced Texture, Shape Features, and Random Forest Classifier with Optimized Particle Swarm Optimization", *International Journal of Biomedical Imaging*, 2022(1), 3211793.

Sukhia, K. N., Riaz, M. M., Ghafoor, A., Ali, S. S. & Iltaf, N. (2019). "Content-based histopathological image retrieval using multi-scale and multichannel decoder based LTP", *Biomedical Signal Processing and Control*, 54, 101582.

Sultani, Z. N. & Yousif, S. A. (2018). "Hybrid feature selection based on mutual information and auc for parkinson's disease classification", *Journal of Theoretical and Applied Information Technology*, 96(18), 6053-6063

Swain, M. J. & Ballard, D. H. (1990). "Indexing via color histograms", *[1990] Proceedings Third International Conference on Computer Vision*, 390-393.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). "Going deeper with convolutions", in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1-9.

Takala, V., Ahonen, T. & Pietikäinen, M. (2005). "Block-based methods for image retrieval using local binary patterns", in *Scandinavian conference on image analysis*. Springer, 882-891.

Tamura, H., Mori, S. & Yamawaki, T. (1978). "Textural features corresponding to visual perception", *IEEE Transactions on Systems, man, and cybernetics*, 8(6), 460-473.

Tan, X. & Triggs, B. (2007). "Enhanced local texture feature sets for face recognition under difficult lighting conditions", in *International workshop on analysis and modeling of faces and gestures*. Springer, 168-182.

Tariq, H., Sheikh, S., Ali, U. M. & Ali, A. (2019). "Domain Specific Content Based Image Retrieval (CBIR) for Feminine Textile Designs", *IJCSNS*, 19(2), 88.

Thirumalaisamy, S., Thangavilou, K., Rajadurai, H., Saidani, O., Alturki, N., Mathivanan, S. k., Jayagopal, P. & Gochhait, S. (2023). "Breast Cancer Classification Using Synthesized Deep Learning Model with Metaheuristic Optimization Algorithm", *Diagnostics*, 13(18), 2925.

Truong, H. P., Nguyen, T. P. & Kim, Y.-G. (2022). "Weighted statistical binary patterns for facial feature representation", *Applied Intelligence*, 52(2), 1893-1912.

Tryon, R. C. (1939). "Cluster analysis. Edwards Brothers", *Ann Arbor, Michigan*, 122.

Tsai, M.-H., Chan, Y.-K., Wang, J.-S., Guo, S.-W. & Wu, J.-L. (2009). "Color-texture-based image retrieval system using Gaussian Markov random field model", *Mathematical Problems in Engineering*, 2009(1), 410243.

Tsai, M.-J. & Tao, Y.-H. (2021). "Deep learning techniques for the classification of colorectal cancer tissue", *Electronics*, 10(14), 1662.

Turhal, U., Günay Yılmaz, A. & Nabiyev, V. (2024). "A new face presentation attack detection method based on face-weighted multi-color multi-level texture features", *The Visual Computer*, 40(3), 1537-1552.

Tyagi, V. (2017). "*Content-Based Image Retrieval: Ideas, Influences, and Current Trends*". Singapore Pte Ltd.

Udousoro, I. C. (2020). "Machine learning: a review", *Semiconductor Science and Information Devices*, 2(2), 5–14.

Unar, S., Wang, X., Zhang, C. & Wang, C. (2019). "Detected text-based image retrieval approach for textual images", *IET Image Processing*, 13(3), 515-521.

Vamsidhar, E., Kenny, M. J. & Gunna, K. (2017). "Particle swarm optimized feature selection for retrieving compressed medical images", in *2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*. IEEE, 1-6.

van den Broek, E. L., Kok, T., Schouten, T. E. & Vuurpijl, L. G. (2008). "Human-centered content-based image retrieval", in *Human Vision and Electronic Imaging XIII*. SPIE, 6806, 575-586.

Varshney, S., Singh, S., Lakshmi, C. V. & Patvardhan, C. (2024). "Content-based image retrieval of Indian traditional textile motifs using deep feature fusion", *Scientific Reports*, 14(1), 10035.

Vassilieva, N. S. (2009). "Content-based image retrieval methods", *Programming and Computer Software*, 35(3), 158-180.

Vensila, C. & Boyed Wesley, A. (2024). "Multimodal biometrics authentication using extreme learning machine with feature reduction by adaptive particle swarm optimization", *The Visual Computer*, 40(3), 1383-1394.

Vieira, G. S., Fonseca, A. U., Sousa, N. M., Felix, J. P. & Soares, F. (2023). "A novel content-based image retrieval system with feature descriptor integration and accuracy noise reduction", *Expert Systems with Applications*, 232, 120774.

Viet Tran, L. (2003). "*Efficient image retrieval with statistical color descriptors*" Ph.D thesis. Linköping University Electronic Press.

Vijaykumar, B. (2020). "*Content Based Image Retrieval Using Features from Deep Convolutional Neural Networks*" Ph.D thesis. Jain University.

Vimina, E. & Divya, M. (2020). "Maximal multi-channel local binary pattern with colour information for CBIR", *Multimedia Tools and Applications*, 79(35), 25357-25377.

Wadhera, A. & Agarwal, M. (2022). "Low dimensional multi-block neighborhood combination pattern for biomedical image retrieval", *Multimedia Tools and Applications*, 81(19), 27853-27877.

Wang, H. H., Mohamad, D. & Ismail, N. (2010). "Semantic gap in cbir: Automatic objects spatial relationships semantic extraction and representation", *International Journal of Image Processing*, 4(3), 192.

Wang, J. Z., Li, J. & Wiederhold, G. (2001). "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries", *IEEE Transactions on pattern analysis and machine intelligence*, 23(9), 947-963.

Wang, M., Li, H., Tao, D., Lu, K. & Wu, X. (2012). "Multimodal graph-based reranking for web image search", *IEEE transactions on image processing*, 21(11), 4649-4661.

Wei, X., Wang, H., Guo, G. & Wan, H. (2014). "A general weighted multi-scale method for improving LBP for face recognition", in *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services: 8th International Conference, UCAmI 2014, Belfast, UK, December 2-5, 2014. Proceedings 8*. Springer, 532-539.

Wei, Z. & Liu, G.-H. (2020). "Image retrieval using the intensity variation descriptor", *Mathematical problems in engineering*, 2020(1), 6283987.

Whitney, A. W. (1971). "A direct method of nonparametric measurement selection", *IEEE transactions on computers*, 100(9), 1100-1103.

Wolpert, D. H. & Macready, W. G. (1997). "No free lunch theorems for optimization", *IEEE transactions on evolutionary computation*, 1(1), 67-82.

Xia, Y., Wan, S., Jin, P. & Yue, L. (2013). "Multi-scale local spatial binary patterns for content-based image retrieval", in *International Conference on Active Media Technology*. Springer, 423-432.

Xie, G., Guo, B., Huang, Z., Zheng, Y. & Yan, Y. (2020). "Combination of dominant color descriptor and Hu moments in consistent zone for content based image retrieval", *IEEE Access*, 8, 146284-146299.

Yang, X.-S. & Deb, S. (2009). "Cuckoo search via Lévy flights", in *2009 World congress on nature & biologically inspired computing (NaBIC)*. Ieee, 210-214.

Yao, X., Liu, Y. & Lin, G. (1999). "Evolutionary programming made faster", *IEEE Transactions on Evolutionary computation*, 3(2), 82-102.

Yazdi, M. & Erfankhah, H. (2020). "Multiclass histology image retrieval, classification using riesz transform and local binary pattern features", *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 8(6), 595-607.

Younus, Z. S., Mohamad, D., Saba, T., Alkawaz, M. H., Rehman, A., Al-Rodhaan, M. & Al-Dhelaan, A. (2015). "Content-based image retrieval using PSO and k-means clustering algorithm", *Arabian Journal of Geosciences*, 8, 6211-6224.

Yuan, X., Yu, J., Qin, Z. & Wan, T. (2011). "A SIFT-LBP image retrieval model based on bag of features", in *IEEE international conference on image processing*. 1061-1064.

Zaheer, Y. (2010). "Content-based image retrieval", in *Second International Conference on Digital Image Processing*. SPIE, 7546, 557-566.

Zhan, Z.-H., Zhang, J., Li, Y. & Chung, H. S.-H. (2009). "Adaptive particle swarm optimization", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6), 1362-1381.

Zhang, G., Huang, X., Li, S. Z., Wang, Y. & Wu, X. (2004). "Boosting local binary pattern (LBP)-based face recognition", in *Chinese Conference on Biometric Recognition*. Springer, 179-186.

Zhang, Z., Zheng, L., Piao, Y., Tao, S., Xu, W., Gao, T. & Wu, X. (2022). "Blind remote sensing image deblurring using local binary pattern prior", *Remote Sensing*, 14(5), 1276.

Zhao, G., Ahonen, T., Matas, J. & Pietikainen, M. (2011). "Rotation-invariant image and video description with local binary pattern features", *IEEE transactions on image processing*, 21(4), 1465-1477.

Zhao, T., Lu, J., Zhang, Y. & Xiao, Q. (2008). "Feature selection based on genetic algorithm for cbir", in *2008 Congress on Image and Signal Processing*. IEEE, 2, 495-499.

Zhou, H., Wang, R. & Wang, C. (2008). "A novel extended local-binary-pattern operator for texture analysis", *Information Sciences*, 178(22), 4314-4325.

Zhou, W., Li, H. & Tian, Q. (2017). "Recent advance in content-based image retrieval: A literature survey", *arXiv preprint arXiv:1706.06064*.

Zivkovic, M., Jovanovic, L., Ivanovic, M., Krdzic, A., Bacanin, N. & Strumberger, I. (2022). "Feature selection using modified sine cosine algorithm with covid-19 dataset", *Evolutionary computing and mobile sustainable networks: Proceedings of ICECMSN 2021*Springer, 15-31.

Appendix

Candidate List of Publications

During this research, 3 papers have been published in international conferences. Additionally, 2 papers have been published in scientific journals. The first is available online with volume and issue numbers, and the second is accepted for publication and available in the "Forthcoming" section of the journal′s website.

A. Conference Publications

1. Rahima Boukerma, Bachir Boucheham & Salah Bougueroua, Image Retrieval Based on Dynamic Weighted Patterns, 2nd International Conference on New Technologies of Information and Communication (NTIC), Mila, December, 2022. **https://doi.org/10.1109/NTIC55069.2022.10100501**
2. Rahima Boukerma, Bachir Boucheham & Salah Bougueroua, Significant Feature Dimensionality Reduction for Histopathology Image Retrieval as a Tool for Healthcare Decisions Support, 2023 International Conference on Decision Aid Sciences and Applications (DASA), Annaba, Algeria, 2023. **https://doi.org/10.1109/DASA59624.2023.10286677**
3. Rahima Boukerma, Bachir Boucheham & Salah Bougueroua, Optimized Deep Features for Colon Histology Image Retrieval, 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), El-Oued, Algeria, 2024. **https://doi.org/10.1109/PAIS62114.2024.10541159**

B. Journal Publications

1. Rahima Boukerma, Bachir Boucheham & Salah Bougueroua, Orthogonal opponent colour local binary patterns: a new colour-texture descriptor for content based-image retrieval, International Journal of Computational Vision and Robotics (IJCVR), InderScience, 15(3):351-378, May 2025, **https://doi.org/10.1504/IJCVR.2025.146295**

2. Rahima Boukerma, Bachir Boucheham & Salah Bougueroua, A Pattern selection Scheme for Content-Based Histopathological Image Retrieval: an Effective Tool for Feature Dimensionality Reduction, International Journal of Bioinformatics Research and Applications (IJBRA), InderScience, **https://doi.org/10.1504/IJBRA.2026.10071482**