People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Skikda - 20 Août 1955
Faculty of Sciences
Department of Computer Science

THESIS

Adaptive Techniques for Content Based Image Retrieval (CBIR): Study and Applications

submitted in fulfilment of the requirements
for the degree of 3rd cycle doctorate (LMD system)

Major: Computer Science

Minor: Computation and Cognition of Informatics Systems

Publically defended by:
Safa Hamreras
On April 4, 2021

Examination Committee:

Smaine Mazouzi	Professor	University of Skikda - 20 Août 1955	Chair
Mehdi Boulaiche	MCA	University of Skikda - 20 Août 1955	Examiner
Said Brahimi	MCA	University of Guelma - 08 Mai 1945	Examiner
Bachir Boucheham	Professor	University of Skikda - 20 Août 1955	Supervisor

Acknowledgements

I thankfully acknowledge my Ph.D supervisor Pr.Bachir Boucheham for all his assistance and availability during the preparation of my thesis, he always provided encouragements and consistent guidance that was necessary to the completion of this research.

My acknowledgment is also addressed to Pr.Smaine Mazouzi for accepting to be the chair of the examination committee, Dr.Mehdi Boulaiche and Dr.Said Brahimi for accepting to evaluate my thesis.

As well, I am grateful to LRES (Laboratoire de Recherche en Électronique de Skikda), of which I am a member.

In addition, I would like to express my sincere gratitude to Pr.Ezequiel López-Rubio for accepting me in his laboratory "Inteligencia Computacional y Análisis de Imágenes" in Málaga, Spain, as well as the other lab members Dr.Miguel A. Molina-Cabello and Dr.Rafaela Benítez-Rochel for all their moral and scientific support.

Finally, I wish to thank my parents for all the support they gave me during my educational journey, without them, I would never reach this level.

Dedicated to my family

Contents

List of Figures

List of Tables

ملخـص

لقد أدت الزيادة المعتبرة في مستودعات الصور من حيث محتواها واستخدامها إلى زيادة الحاجة لهندسة أنظمة فعالة للبحث عن الصور. في هذا الصدد، يُعد البحث عن الصور بالاعتماد على خصائصها المرئية واحدا من الأنظمة المطورة.

تعد "الفجوة الدلالية" من أبرز التحديات التي تواجه أنظمة البحث عن الصور بناء على المحتوى. تحدث هذه الفجوة بين تمثيل الصور - أي استخراج خصائصها المرئية - بواسطة نظام البحث المستعمل والمعاني الحقيقية التي تتضمنها هذه الصور. من الجليّ أن يتم القضاء على هذه الفجوة أو على الأقل تقليصها من أجل التقريب بين نظام البحث المستعمل و النظام البصري البشري في تمثيل الصور. في هذا السياق، يجب علينا تصميم أنظمة تأخذ بعين الاعتبار المفهوم البشري لهذه الصور.

في هذه الأطروحة، نعالج هذا التحدي من خلال تسليط الضوء على النموذج الذي فرض نفسه في عديد المجـالات، ألا وهو نموذج التكيُّف. في الواقع، هناك حاجة لإدماج المرونة في أنظمة استرجاع الصور بحيث تقوم بعملية البحث آخذة بعين الاعتبار بعض المعايير، من أبرزها بيانات الإدخال. نركز بشكل خاص على إدماج التكيف في مرحلة تمثيل الصور، حيث نهدف إلى تضييق الفجوة الدلالية. يتم إدماج هذا التكيف إما عن طريق اختيار خوارزمية تمثيل الصور أو ضبط متغيراتها في مرحلة التدريب، من أجل زيادة فاعليتها على المهمة المستهدفة.

أولا، في مساهمتنا الأولى، نقوم بتطبيق نموذج لاختيار الخوارزمية الأكثر فاعلية لتمثيل كل صورة على حدة. في هذا العمل، تم تحليل خوارزمية نظام البحث عن الصور إلى العديد من المتغيرات، حيث تم اختيار متغيرة تمثيل الصورة بناءً على الصورة المُدخلة. في مساهمتنا الثانية، نقوم باستعمال الخوارزمية المسماة "الشبكة العصبية التلافيفية" في مرحلة تمثيل الصور، حيث قمنا أولا بضبط متغيرات هذه الخوارزمية في مرحلة التدريب. بعد ذلك، تم استخراج تمثيلات هده الصور باستعمال هذه الخوارزمية. أخيرا، تعتمد مساهمتنا الأساسية على استعمال عدة شبكات عصبية تلافيفية في تمثيل هذه الصور، حيث تم استخراج تمثيلات الصور باستعمال كل شبكة على حدة، ليتم دمج هده التمثيلات من أجل زيادة فاعليتها.

في الأخير، تجدر الإشارة إلى أن كل مساهماتنا أدت إلى تحسن معتبر في أداء الأنظمة المُطورة للبحث عن الصور بناء على المحتوى، مما يدل على تضييق الفجوة الدلالية المذكورة أعلاه، و هي الغاية الأسمى من هذا البحث.

الكلمات المفتاحية: البحث عن الصور بناءً على المحتوى، نموذج التكيف، الفجوة الدلالية، نظرية لا غذاء مجاني، نموذج الاختيار، اختيار الخوارزمية، الشبكات العصبية التلافيفية، التعلم الجماعي، الذكاء الحسابي.

Abstract

The significant growth of images repositories in terms of content and usage has increased the need of conceiving powerful image retrieval systems. In this regard, CBIR is the process of searching images given a query based on their visual aspects.

One of the most serious challenges that is facing CBIR systems is "the semantic gap". This gap occurs between images representation given by a CBIR system and the true semantics included within these images. The semantic gap should be bridged or at least reduced in order to bring closer the low level vision of the CBIR system and the high level vision of the Human Visual System (HVS). This could be achieved through designing CBIR systems that behave with respect to the human understanding of images.

In this thesis, we address this challenge by shedding light on a paradigm that has already imposed itself in many fields, that is the adaptation paradigm. Indeed, there is a need to inject a flexibility in the designed CBIR systems, so that they behave depending on some criterion, mainly the input data. We particularly integrate the adaptation into images characterization phase so as to reduce the semantic gap. This adaptation is carried out by selecting the characterization algorithm or tuning its parameters in a training phase, in order to increase its performance with respect to the target task.

First, we start with the selection paradigm, where we apply the algorithm selection based on RICE model to select the best performing CBIR-algorithm for each image. In this work, the CBIR-algorithm was decomposed to many variants, where the feature variant was selected based on the input image. In the second contribution, we propose to use a Convolutional Neural Network (CNN) in the feature extraction stage. This latter is first finetuned on the target images dataset in order to adapt its parameters. The resulting network is then used to extract the class probability vectors from input images to use them as representations. Last, our main contribution consists of using an ensemble of CNNs that collaborate to extract relevant features from images. Herein, the class probability vectors are extracted from images using each ensemble member. They are then combined to make a powerful image representation.

Finally, it is noteworthy that all our contributions have led to a significant improvement in the performance of the developed CBIR systems. This latter is a relevant indicator of narrowing the semantic gap within these systems, which is the ultimate goal of this thesis.

Keywords: Content based image retrieval, Adaptation paradigm, Semantic gap, No free lunch theorem, Selection paradigm, Algorithm selection, Convolutional neural networks, Ensemble learning, Computational intelligence.

Résumé

La croissance significative des dépôts d'images en termes de contenu et utilisation a poussé le besoin de concevoir des systèmes de recherche d'images puissants. À cet égard, CBIR est le processus de la recherche d'images à partir d'une requête en se basant sur leurs aspects visuels.

L'un des défis les plus sérieux auxquels les systèmes CBIR sont confrontés est "le fossé sémantique". Ce fossé se produit entre la représentation des images donnée par un système CBIR et la véritable sémantique incluse dans ces images. Le fossé sémantique doit être comblé ou au moins réduit afin de rapprocher la vision de bas niveau du système CBIR et la vision de haut niveau du système visuel humain (HVS). Ceci pourrait être réalisé en concevant des systèmes CBIR qui se comportent selon la compréhension humaine des images.

Dans cette thèse, nous adressons ce défi en mettant en lumière un paradigme qui s'est déjà imposé dans de nombreux domaines, c'est le paradigme de l'adaptation. En effet, il est nécessaire d'injecter une flexibilité dans les systèmes CBIR conçus afin qu'ils se comportent en fonction de certains critères, notamment les données d'entrée. Nous intégrons particulièrement l'adaptation dans la phase de la caractérisation des images afin de réduire le fossé sémantique. Cette adaptation est réalisée en choisissant l'algorithme de caractérisation ou en ajustant ses paramètres, d'une manière à améliorer sa performance par rapport à la tâche cible.

Tout d'abord, nous commençons par le paradigme de sélection, où nous appliquons la sélection d'algorithme basée sur le modèle de RICE, afin de sélectionner l'algorithme CBIR le plus performant pour chaque image. Dans ce travail, l'algorithme CBIR a été décomposé en de nombreuses variantes, où la variante de la caractéristique a été sélectionnée en fonction de l'image en entrée. Dans la deuxième contribution, nous proposons d'utiliser un réseau de neurones convolutif (CNN) dans l'étape de l'extraction des caractéristiques. Ce dernier est d'abord affiné par rapport à la base d'images cible afin d'adapter ses paramètres. Le réseau résultant est ensuite utilisé pour extraire les vecteurs de probabilité de classe à partir des images saisies, afin de les utiliser comme représentations. Enfin, notre contribution principale consiste à utiliser un ensemble de CNNs qui collaborent pour extraire des caractéristiques pertinentes des images. Ici, les vecteurs de probabilités de classes sont extraits à partir des images en utilisant chaque membre de l'ensemble. Ils sont ensuite combinés pour former une représentation puissante des images.

Enfin, il convient de noter que toutes nos contributions ont conduit à une amélioration significative de la performance des systèmes CBIR développés. Ceci est un indicateur pertinent de la réduction du fossé sémantique au sein de ces systèmes, qui est le but ultime de cette thèse.

Mots clés: Recherche d'images basée sur le contenu, Paradigme d'adaptation, Fossé sémantique, Théorème du no free lunch, Paradigme de sélection, Sélection d'algorithme, Réseaux de neurones convolutifs, Apprentissage ensembliste, Intelligence computationnelle.

Abbreviations

ACO	Ant Colony Optimization
AI	Artificial Intelligence
ANN	Artificial Neural Network
AQE	Average Query Expansion
AS	Algorithm Selection
BOW	Bag Of visual Words
CBIR	Content Based Image Retrieval
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DS	Distance Selection
DT	Decision Tree
EC	Evolutionary Computing
FL	Fuzzy Logic
FS	Feature Selection
FV	Fisher Vector
GA	Genetic Algorithm
GAN	Generative Adversarial Network
GLCM	Gray Level Co-occurrence Matrix
GPU	Graphical Processing Unit
HVS	Human Visual System
KNN	K Nearest Neighbour
LBP	Local Binary Pattern
LSTM	Long Short Term Memory
mAP	mean Average Precision

MAS	Multi-Agent System
MCNG	Matrice de Co-ocurrence des Niveaux de Gris
ML	Machine Learning
NFLT	No Free Lunch Theorem
PSO	Particle Swarm Optimization
QE	Query Expansion
RBS	Rule Based System
RF	Relevance Feedback
RNN	Recurrent Neural Network
RNN	Recursive Neural Network
SFS	Sequential Forward Selection
SI	Swarm Intelligence
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
TBIR	Text Based Image Retrieval
TPU	Tensor Processing Unit

Part I

General Introduction

Chapter 1

General introduction

1.1 Context of study

With the new technological era, images have gained a large interest. This interest manifests itself in high-resolution screens and cameras, as well as computing devices designed especially to operate on images, such as scanners. As a result of this interest, there is an ongoing demand for increasing the storage capacity of images repositories, depicted in cloud services aimed at hosting a huge amount of images instantly uploaded by internet users, or images repositories designed for professional purposes. These repositories are utilized through uploading images, browsing them, or searching for particular images sharing the same semantic. Since necessity is the mother of invention, there was a need to develop techniques to search images in those large repositories with aim to end up with accurate retrieval results.

So far, in the literature, image retrieval has been addressed through two main approaches. The first one is Text Based Image Retrieval (TBIR), where images characterization is based on human annotations. This approach of retrieving images is, however, problematic for a number of reasons, particularly, the subjectivity introduced by involving the user in images annotation, and the huge amount of time that could be dedicated to this task, especially with the growing size of images archives. Alternatively, CBIR has been introduced to tackle these issues by retrieving images based on their visual features such as color, texture and shape. These features are then exploited to measure the similarity between an input query and database images using a similarity matching measure.

Due to its popularity among web surfers, there exists a large number of CBIR web engines such as Pixolution [1], Shutterstock [2], TinEye [3], as well as the image search service of the giant Google [4]. Image search based on the visual content is not limited to non professional purposes. Indeed, CBIR has critical professional aspects, particularly in medicine, where involving CBIR in this area was a qualitative step forward to make accurate diagnosis using medical image analysis (Owais et al., 2019), or social security using criminal identification based on sketch queries (Shriram et al., 2015). A wide variety of other applications of CBIR exist, such as remote sensing (Napoletano, 2018), surveillance systems (Ahmad et al., 2018), and face recognition (Fachrurrozi et al., 2017).

The emergence of this area in many fields constitutes another reason behind the need of conceiving advanced techniques. In this regard, two main directions have been proposed, either by designing new algorithms that better solve some specific retrieval tasks (Höschl IV and Flusser, 2016), or designing adaptive techniques that could be tuned depending on the target retrieval task(s) (Fu et al., 2016b). In this thesis, we are interested in the second approach, that is the adaptation paradigm.

Adaptation has emerged in many systems as a paradigm for tuning systems to better meet the requirements of the target task (Truong, 2016) (Wen et al., 2018) (Liu et al., 2018) (Gomes et al., 2017) (Li et al., 2019). In addition, it has already made its mark in the CBIR area through an immense range of techniques such as selection based techniques, where most of them attempt to select the best features subset to characterize images (Belattar et al., 2018), evolutionary computing techniques (Alsmadi, 2017), or machine learning based techniques (Saritha et al., 2019). This inspired us to follow this direction with a view to put our mark on it.

1.2 Problematic and motivation

The great variety of images in terms of their visual content has raised a problematic question: How to maintain the good outcome of a CBIR system in the shadow of all the content diversity between images?

[1] `https://pixolution.org/`
[2] `https://www.shutterstock.com/`
[3] `https://tineye.com/`
[4] `https://images.google.com/`

In order to answer this question, we should invoke the No Free Lunch Theorem (Ho and Pepyne, 2002), on which our thesis stands. This theorem states that there no single best strategy, i.e. it is not possible to develop a strategy that outperforms all others on all possible problem instances. If we apply this theorem on the CBIR framework, then the strategies consist in the CBIR system components, such as the used characterization feature or the similarity matching measure, and the problem instances consist in the retrieval tasks.

Standing on this theorem, it is not possible to develop a CBIR system with fixed components that could be the best performing on all possible retrieval tasks. Consequently, there was a need to conceive systems that change their behaviour depending on the problem at hand. This change in behaviour manifests itself in the component algorithms or their parameters, resulting in a system that adapts its outcome with respect to the retrieval task, namely an adaptive system.

In the CBIR area, we have noticed that the adaptation could be carried out with respect to two different criterion, the used dataset (Rashno and Sadri, 2017) or the query image (Benloucif and Boucheham, 2014). For the first criterion, the system is adapted in an offline phase, i.e. a training phase, where the component algorithms are tuned and fixed to be used for all input queries. In addition, in the query based approach, the system changes its behaviour depending on the query where the adaptation is carried out online. In order to give a more complete study of adaptation within CBIR, we explore both adaptation approaches.

1.3 Objectives

The objectives behind this study could be summarized in three main points:

- A review of basic CBIR techniques and the related concepts in order to clarify the study area.
- A review of adaptive techniques in the literature in order to emphasize the emergence of adaptation in many areas, as well as showing its role in conceiving powerful systems.
- Study of adaptation in CBIR through reviewing the existing adaptive techniques as well as the suggestion of new ones. These techniques have in common the purpose of empowering CBIR systems through reducing the semantic gap within these systems.

1.4 Published content and contributions

In order to achieve the objectives above, we have contributed with three research items:

Adaptive Content Based Image Retrieval based on RICE Algorithm Selection Model (Hamreras and Boucheham, 2018) This was our entry to the adaptation paradigm in CBIR. In this paper, we applied RICE model for algorithm selection to tackle a main problematic in CBIR, which is image characterization. The popularity of this model in many fields and the poor interest in it in the CBIR area have inspired us to utilize it in our research. The idea was to tune the color space and feature components of a CBIR-algorithm, depending on each query image. These algorithms are selected based on their performance for each query, using a KNN search in a training dataset. This dataset includes tuples of images samples and their respective selected algorithms, therefore, the adaptation here is query based.

Content Based Image Retrieval by Convolutional Neural Networks (Hamreras et al., 2019) We used CNNs in our study due to their recent rousing success in imaging fields, including CBIR. The idea was to finetune a pretrained CNN to adapt it to the target dataset, then to use this CNN to extract features from images. Since the CNN parameters are fixed in the training phase, the adaptation is dataset based. The proposed technique was simple, but it led to satisfactory results and outperformed the compared techniques on the experimented dataset.

Content Based Image Retrieval by Ensembles of Deep Learning Object Classifiers (Hamreras et al., 2020) This constitutes the main contribution of our thesis. In this work, an extension of the previous work has been made by combining several CNNs that collaborate to extract relevant features from images, namely an ensemble of CNNs. It should be highlighted that ensemble learning is not popular in CBIR, and compared to the few existing works, the proposed architecture is novel. Two approaches for ensemble learning have been considered: the use of different CNNs architectures trained on the same target task, or the same CNN architecture finetuned on different training subsets using the bagging technique. We conducted an extensive study to evaluate the best parameters of the proposed architecture depending on the target dataset. Therefore, the adaptation is dataset based. In addition, we proposed a query expansion technique for our architecture

to enhance the retrieval results, which means that a query based adaptation was also carried out. Our proposed technique outperformed the recent literature that we could collect on the used dataset.

1.5 Thesis outline

The remainder of this thesis is organized in three parts. Part II is dedicated to the state of the art of CBIR and adaptation, as follows:

Chapter 2 This chapter is an overview of the CBIR area. We give the general workflow of CBIR systems. Particularly, we emphasize the main components of these systems including feature extraction and similarity matching measures. At the end of this chapter, the reader could be familiar with this area and have a better understanding of CBIR systems.

Chapter 3 This chapter lines up the different aspects of adaptation depicted in a variety of techniques, as well as showing its role in conceiving powerful systems.

Chapter 4 This chapter is the heart of our thesis. Herein, we discuss the adaptation in the CBIR area by explaining how this paradigm has been addressed in the literature.

Part III is dedicated to personal contributions. In each chapter, we explain the used paradigm, give the methodology and the conducted experiments, as well as the obtained results.

Chapter 5 Our first contribution: "Adaptive Content Based Image Retrieval based on RICE Algorithm Selection Model".

Chapter 6 Our second contribution: "Content Based Image Retrieval by Convolutional Neural Networks".

Chapter 7 Our last and main contribution: "Content Based Image Retrieval by Ensembles of Deep Learning Object Classifiers".

We conclude with general conclusions and perspectives to discuss and evaluate the main findings of our research, as well as spotting the light on the potential future works.

Part II

State of the Art

Chapter 2

State of the art of CBIR

Over time, CBIR has known a significant growth in terms of the addressing techniques, making it a heavily studied research area. In this first chapter we will overview this area to present the main parts of the CBIR framework as well as the related concepts. Additionally, we will shed the light onto the most ubiquitous basic techniques and concepts in CBIR in order to give a clear vision of it.

2.1 The general scheme for CBIR systems

As shown in Figure 2.1, a CBIR system follows two main steps:

Indexation phase (offline) Where all database images features are extracted and stored. These features are used as images descriptors and they can represent a wide content diversity included within images, such as color, texture, and shape. In addition, these features should be "discriminant", i.e. they should reflect the diversity in images in terms of the included semantical concepts, so as to reduce the semantic gap, which leads to a better distinction between similar and dissimilar images.

Retrieval phase (online) The aim of this step is to extract the most similar images to an input query image. Query image features are first computed with respect to the previous step, i.e. the query features are the same as the database features, so that a comparison could be carried out between them. Several techniques could be used to compare between images features, where the most popular are distance measures and similarity metrics,

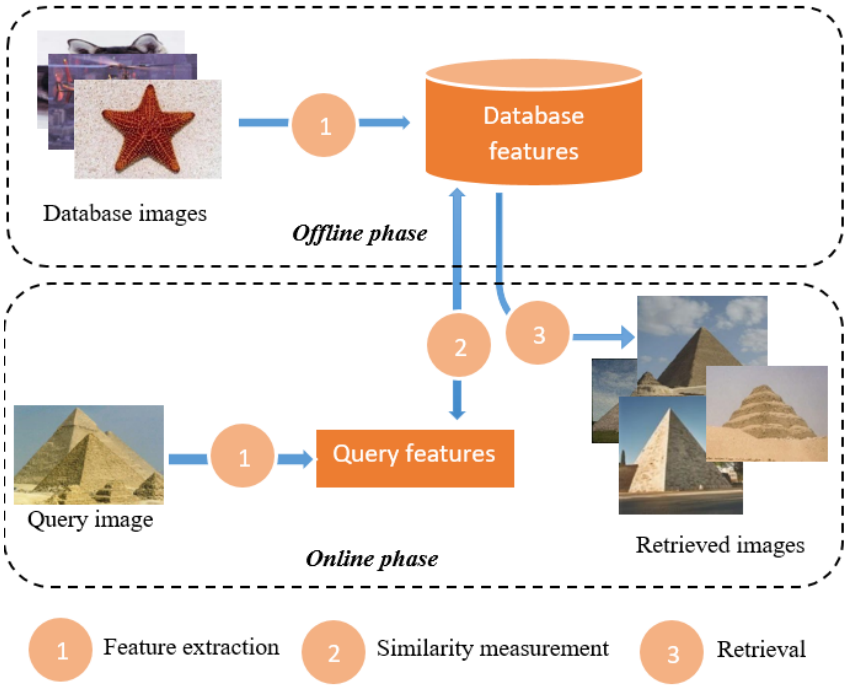


FIGURE 2.1: The workflow of CBIR systems including feature extraction, similarity measurement, and retrieval phases.

that will be discussed in this chapter. Based on this comparison, database images could then be ranked depending on their relevance to the query.

2.2 Explicit feature extraction techniques

Explicit feature extraction techniques operate on images in a transparent way to the user, so that the results are easily interpretable and the user can understand the generated results. These techniques range from color, texture, shape and keypoints detectors.

Even though these features date back to the nineties, they are still used due to their robustness in images characterization. In this section, we introduce each of them and highlight the commonly used algorithms in each technique.

2.2.1 Color based features

Color is defined as the result of the incidence of light on an object (Poynton, 1997). It is a very rich component in terms of the values it can take. Indeed, the human eye can distinguish around 10 million colors through its retina. The quantity of colors contained in digital images

often reflect its quality, i.e. images with high resolution contain more pixels, and thus can include more colors. These colors are encoded as arrays, where each cell refers to a pixel of an image and takes as value the pixel's color (Poynton, 1997). In this subsection, we discuss the commonly used color spaces within CBIR as well as some of the used algorithms to encode this component.

2.2.1.1 Color spaces

A color space is defined by its channels, with each having a specific range of values to represent a color property. Colors are produced by combining these channels values, where different combinations result in different colors. The existence of several color spaces produced different ways, i.e. algorithms for color representation, where the color space is an input parameter to the algorithm, resulting in algorithms with different parameter settings. Well known color spaces used within CBIR are discussed below.

RGB color space This color space is used to encode colors in digital devices, such as TVs and computer screens. It has three channels: Red, Green, and Blue, where each channel values range from 0 to 255, resulting in 16,777,216 million colors that a digital image pixel can take.

A geometric representation of this color space is shown in Figure 2.2.

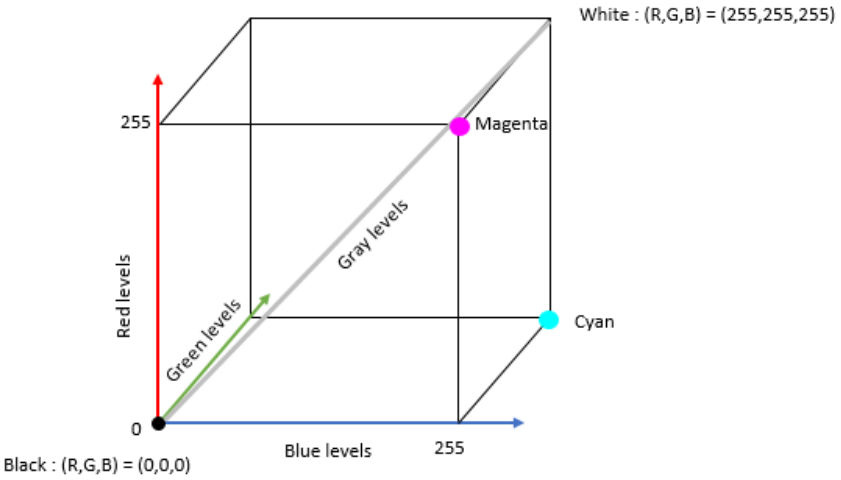


FIGURE 2.2: RGB color space cube.

HSV HSV stands for Hue, Saturation, and Value, which are the three components of this color space. The hue indicates the color, the saturation indicates its purity, and the value indicates the luminance of the color. Figure 2.3 shows HSV cylinder representation.

Since HSV is based on RGB color space, it is possible to convert an image from RGB color space to HSV color space, as follows (Loesdau et al., 2014):

$$R, G, B \in [0, 1],\ Max = max[R, G, B],\ MIN = min[R, G, B] \tag{2.1}$$

$$H = \begin{cases} 0, if R = G = B \\ 60° \times (\frac{G-B}{MAX-MIN}), if\ MAX = R \\ 60° \times (2 + \frac{B-R}{MAX-MIN}), if\ MAX = G \\ 60° \times (4 + \frac{R-G}{MAX-MIN}), if\ MAX = B \end{cases} \tag{2.2}$$

$$S = \begin{cases} 0, if\ R = G = B \\ \frac{MAX-MIN}{MAX},\ else \end{cases} \tag{2.3}$$

$$V = MAX \tag{2.4}$$

Unlike RGB, HSV is a correlated color space. Figure 2.3 shows a representation of this space.

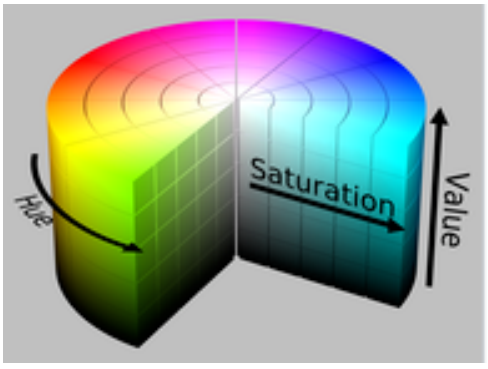


FIGURE 2.3: HSV color space cylinder [1].

CIE XYZ This color space was defined by CIE (Commission Internationale de l'Eclairage). A linear transformation from RGB color space components allows to easily define XYZ primaries X,Y, and Z (Acharya, 2002), which are given in the following equations (Judd, 1970):

$$X = \int_0^\infty E_\lambda \beta_\lambda \bar{x}(\lambda) d\lambda \tag{2.5}$$

[1] `https://en.wikipedia.org/wiki/HSL_and_HSV/`

$$Y = \int_0^\infty E_\lambda \beta_\lambda \bar{y}(\lambda) d\lambda \tag{2.6}$$

$$Z = \int_0^\infty E_\lambda \beta_\lambda \bar{z}(\lambda) d\lambda \tag{2.7}$$

Where $\bar{x}(\lambda)$,$\bar{y}(\lambda)$,$\bar{z}(\lambda)$ refer to the red, blue, and green sensitivities of the human eye, β_λ is a specimen of spectral radiance factor, and E_λ is the corresponding light source of spectral distribution. Figure 2.4 shows the chromacity diagram of this color space.

CIE XYZ is composed of spectral curves called "standard observer" (Wyman et al., 2013) as shown in Figure 2.4.

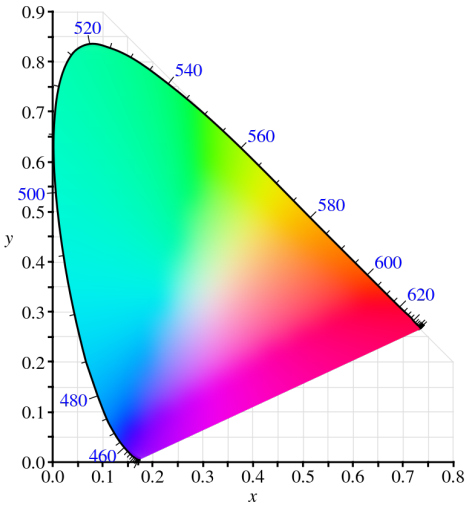


FIGURE 2.4: XYZ color space curve [2].

The conversion from RGB color space to CIE LAB is given as follows (Yang et al., 2010):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.201 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.117 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.8}$$

CIE Lab Similarly to CIE XYZ, CIE Lab was defined by CIE. In this color space, the L channel ranges from 0 to 100 and represents degrees from black to white, the a and b channels range from -128 to 127 and represent degrees from red to green and yellow to blue, respectively (Murali and Govindan, 2013).

It is possible to convert from CIE XYZ to CIE Lab using the following transformation (Rautiainen et al., 2001):

[2] `https://fr.wikipedia.org/wiki/CIE_XYZ/`

$$L = 116 \times f(\frac{Y}{Y_n}) - 16 \tag{2.9}$$

$$a = 500 \times (f(\frac{X}{X_n}) - f(\frac{Y}{Y_n})) \tag{2.10}$$

$$b = 200 \times (f(\frac{Y}{Y_n}) - f(\frac{Z}{Z_n})) \tag{2.11}$$

Where $X_n = 0.31006$, $Y_n = 0.31616$, and $Z_n = 0.37378$.

$$f(t) = \begin{cases} t^3, if\ t > 0.008856 \\ 7.787t + \frac{16}{116},\ else \end{cases} \tag{2.12}$$

Figure 2.5 shows the CIE Lab cylinder representation.

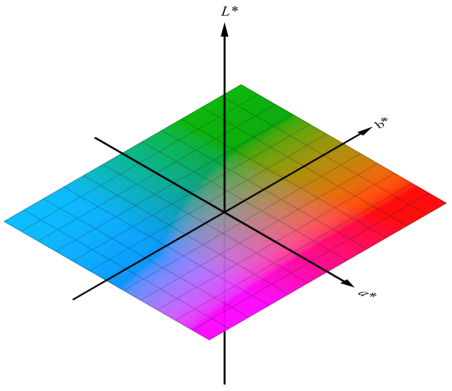


FIGURE 2.5: CIE Lab color space representation [3].

CIE Luv CIE Luv is another color space defined by CIE. This color space is uniform, however, it may significantly suffer from noise when it is transcripted from RGB color space. The reason is that the transcription from RGB to CIE Luv is highly non linear, which makes this color space highly sensitive to noise (Shafarenko et al., 1998).

Figure 2.6 shows CIE Luv cylinder representation.

The conversion from XYZ to Luv is carried out as follows (Xiang et al., 2009):

$$u\prime = \frac{4X}{X + 15Y + Z} \tag{2.13}$$

[3] `https://fr.wikipedia.org/wiki/L*a*b*_CIE_1976/`
[4] `https://en.wikipedia.org/wiki/CIELUV/`

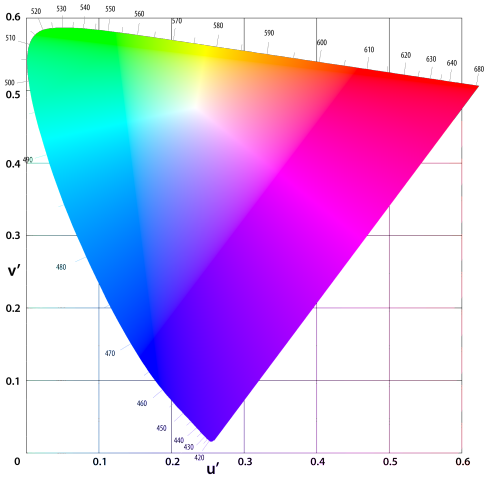


FIGURE 2.6: CIE Luv color space curve [4].

$$v\prime = \frac{9X}{X + 15Y + Z} \tag{2.14}$$

If the reference luminance of white light is considered, X,Y, and Z are denoted as X_n, Y_n, and Z_n, respectively.

$$U = 13L(u\prime - u\prime_n) \tag{2.15}$$

$$V = 13L(v\prime - v\prime_n) \tag{2.16}$$

$$L = 116(\frac{Y}{Y_n})^{\frac{1}{3}} - 16, \frac{Y}{Y_n} \leq 0.08856 \tag{2.17}$$

In order to better compare between color spaces, Table 2.1 recapitulates the properties of each color space. The considered properties are defined as follows:

linearity A color space is linear when the correlation between the color intensity value and the its perceived intensity is linear. Linear color spaces could be derived from the RGB color space using a linear transformation (Yang et al., 2010).

correlation In a correlated color space, changing a color in a coherent way requires changing all color channels, which complicates the production of a new similar color (Reinhard et al., 2001).

device dependency In device dependent color spaces, the same channels combination may result in different colors depending on the display device.

uniformity In a uniform color space, the difference between the produced colors is the same perceived by humans.

TABLE 2.1: Colors spaces properties.

Color space	Correlation	Uniformity	Linearity	Device-dependency
RGB	Yes	No	No	Yes
HSV	Yes	Yes	No	Yes
XYZ	Yes	No	Yes	No
CIE Lab	Yes	Yes	No	No
CIE Luv	Yes	Yes	No	No

2.2.1.2 Color algorithms

Depending on the used color space and algorithm, the color can be extracted and represented in several ways. In this subsection, we describe some of the most used algorithms in color characterization.

Color histogram The idea of color histograms (Swain and Ballard, 1991) is to create color intervals so that each pixel value belongs to one of the intervals, called bins. Each bin is assigned a frequency to indicate how many pixels have one of the bin colors. The bin size affects the histogram dimensionality, where lower number of bins results in a lower dimensional histogram. According to Chakravarti and Meng (2009), the advantage of using color histograms is their ability to retrieve similar images regardless of their positions, orientation, and size, however, one of its main limitations is its inability to capture the spatial information.

Color moments The idea of color moments is to transform the color histogram to a reduced set of features. In other words, these moments represent the color distribution in an image based on its three main features (Stricker and Orengo, 1995), that is: the mean E_i, the standard deviation σ_i, and the skewness s_i, which represent respectively, the average color in the image, the deviation and the skewness of each color channel. These moments are given in the following equations (Stricker and Orengo, 1995):

$$E_i = \sum_{j=1}^{N} \frac{1}{N} P_{ij} \tag{2.18}$$

$$\sigma_i = (\frac{1}{N}\sum_{j=1}^{N}(P_{ij} - E_i)^2)^{\frac{1}{2}} \tag{2.19}$$

$$s_i = (\frac{1}{N}\sum_{j=1}^{N}(P_{ij} - E_i)^3)^{\frac{1}{3}} \tag{2.20}$$

Color coherence vector (CCV) The motivation behind CCV is to inject the spatial coherence information in the color histogram, resulting in a more powerful representation (Huang et al., 1999). CCV classifies images pixels into coherent/non coherent depending on their belonging to a large region having the same color. Thus, the coherent regions are more sizable and have more importance in images representation. These values are then memorized in a single feature vector, where each color is assigned the number of its coherent and non coherent pixels (Pass et al., 1997).

2.2.2 Texture based features

The great variety of patterns in images is behind the difficulty of drawing up a clear definition for texture property (Arivazhagan and Ganesan, 2003). However, in the literature, the texture is generally defined as the repetition of patterns within an image (Emerson et al., 1999) (Arivazhagan and Ganesan, 2003) (Srinivasan and Shobha, 2008) (Wei et al., 2009) (Backes et al., 2011).

Texture can be classified into two major categories (Raju et al., 2010):

Structural Where the concept is to find out how texture elements are arranged with respect to a placement rule (Raju et al., 2010) (Srinivasan and Shobha, 2008). These textures are applicable to regular textures (Humeau-Heurtier, 2019).

Statistical Where the idea is to extract a set of statistics by computing local features which consist in the spatial distribution of gray values (Srinivasan and Shobha, 2008) (Van Gool et al., 1985). The well-known Gray Level Co-occurence Matrix (GLCM) algorithm belongs to this category.

There exists several texture extraction algorithms, some of the most used in the CBIR context are:

LBP This algorithm computes the feature vector in an iterative process, where a predefined number of neighbour pixels are considered to compute the local binary pattern of each pixel. Ojala et al. (2002) beautifully explains this algorithm: "The name 'Local Binary Pattern' reflects the functionality of the operator, i.e. a local neighborhood is thresholded at the gray value of the center pixel into a binary pattern". The flexibility of this method is behind its popularity (Pietikäinen and Zhao, 2015). Indeed, there exists several variants of this method, including LTP (Tan and Triggs, 2010), CLBP (Guo et al., 2010), VLBP(Zhao and Pietikainen, 2007), tLBP (Trefnỳ and Matas, 2010), dLBP (Trefnỳ and Matas, 2010), and GLIBP (Bougueroua and Boucheham, 2018). According to Humeau-Heurtier (2019) LBP has the advantage of belonging to both statistical and structural techniques, which increases its performance, however, some of its drawbacks are sensitivity to image rotation, noise, and blurriness. Moreover, the extracted information depends on the neighborhood size, which may not be enough if the latter is small.

GLCM The Gray Level Co-occurence Matrix is based on the same concept of LBP, i.e. computing the features in the pixel neighbouring space. the GLCM captures the spatial relationships between pixels (Hafizah et al., 2012) and stores them in a matrix. It suffers from high dimensionality, however, it is easy to compute and proved to be effective in many applications (Humeau-Heurtier, 2019).

MCNG The Gray Level Co-occurence Matrix of Haralick describes the spatial interrelationships of the gray tones in a texture (Haralick, 1979), depending on a defined distance or angular spatial relationships (Bougueroua and Boucheham, 2017). The advantage of this technique is the invariance to monotonic gray transformation, however, it does not capture the tonal primitives shapes (Haralick, 1979).

Gabor filters This algorithm is inspired by the visual cortex (Turner, 1986). It is based on modeling filters that are sensitive to different frequency bands and orientations (Guérin-Dugué and Palagi, 1994). These filters are known to be robust against noise (Andrysiak and Choraś, 2005) and provide a powerful image representation, however, they may suffer from redundancy at different scales (Humeau-Heurtier, 2019).

2.2.3 Shape based features

In order to achieve a good retrieval accuracy, a shape descriptor should recognize perceptually similar images (Mingqiang et al., 2008). The change in visual aspects of an image such as

rotation and scale should not greatly affect its visual appearence; consequently, there is a need to design features that are invariant to these visual aspects in order to effectively represent and retrieve images (Li et al., 2008). In this regard, shape features are known to be invariant to rotation, translation, and scale (Andaló et al., 2010), making them well suited for the retrieval task.

Several requirements should be met when designing shape features. According to Mingqiang et al. (2008), a good shape descriptor should be as complete as possible to cover the necessary visual aspects, compact regarding the representation and storage, and it should ensure a simple distance computation to reduce the computational cost.

Shape features can be classified into two categories:

Countour based features In this category, features are extracted from images boundaries. This approach has the drawback of ignoring inner important patterns (Li et al., 2018), however, it is efficient for shapes having complex boundaries (Liu et al., 2007). Algorithms examples are: Fourier descriptors (Persoon and Fu, 1977), Elastic matching (Younes, 1999), and Wavelet descriptors (Shen and Ip, 1999).

Region based features This approach gives account to the inner information within a shape, however, it requires a more computational cost, due to the need of selecting relevant keypoints for the comparison (Li et al., 2018). Some well-known algorithms of this category are: Legendre moments (Hu, 1962), Zernike moments (Teh and Chin, 1988), and Geometric moments (Teague, 1980).

2.2.4 Keypoint based techniques

Semantically similar images may differ in terms of some visual aspects that are not related to their true semantics. These visual aspects may be considered by low level features, which would badly affect the image representation, and hence the retrieval process. In this regard, keypoint based techniques aim to extract images regions that are invariant to those visual properties in order to provide a better visual characterization. Related algorithms are presented below.

2.2.4.1 Scale Invariant Feature Transform (SIFT)

SIFT features are composed of local feature vectors that are invariant to translation, rotation, scaling, and partially invariant to illumination changes and affine or 3D projection (Lowe, 1999). Below, is the workflow of SIFT features extraction (Lowe, 2004).

Scale space extrema detection The role of this step is to detect potential keypoints that are invariant to rotation and scale, by searching over all image rotations and scales. The keys are located at minima and maxima of a difference of Gaussian function as follows (Lowe, 1999):

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \tag{2.21}$$

Where $\sigma = sqrt2$.

Key point localization In this step, only the most stable key localizations are selected.

Orientation assignment Herein, invariance to rotation is achieved (Bicego et al., 2006) by assigning one or several orientations to the selected keypoints.

Keypoint descriptor This final step adds the invariance to illumination to the already selected keypoints.

According to Lowe (1999), the recognition using SIFT features is referred to as "feature matching", where, for each keypoint descriptor, the nearest neighbours are identified using the Euclidean distance, and the probability that a match is correct is then computed as the difference between the distances of two subsequent neighbours, that is the distance ratio. A match is considered false if the distance ratio exceeds a predefined threshold.

The robustness of SIFT features is back to their invariance to the change of the above-mentioned image visual properties. However, their major drawback is the high dimensionality which increases the computational cost, especially in the matching phase (Alhwarin et al., 2010).

2.2.4.2 Bag Of visual Words (BoW)

The idea behind BoW is to reduce local images decsriptors by creating a vocabulary of visual words, where a visual word is a selected image descriptor meant to be used in image characterization. These features are obtained as follows (Tsai, 2012):

- Detect and compute local descriptors, generally SIFT features.
- Use a clustering algorithm such as k-means to quantize the extracted features into a vocabulary of visual words.
- For each image, the bag of visual words is a histogram depicting the number of visual words in this image (Csurka et al., 2004)

Since these features are based on SIFT like features, they inherit the invariance property from them. However, contrary to SIFT features, they ensure a lower computational cost (Csurka et al., 2004) thanks to the reduction of features size through building a vocabulary of descriptors.

2.2.4.3 Fisher vector (FV)

Fisher vector is the result of the aggregation of patch features into a single high dimensional vector (Simonyan et al., 2013).

The construction of the FV is based on fitting a probability density function modeled by the Gaussian Mixture Model (GMM) to the local features (Peng et al., 2014). The derivatives of the log-likelihood of the GMM are encoded with respect to the GMM parameters (Simonyan et al., 2013) to be aggregated into a single vector (Peng et al., 2014).

Compared to BoW, the main drawback of this representation is its high dimensionality, which makes it unsuited for large scale applications, such as large scale image retrieval. However, compared to the same algorithm, it can be computed for smaller vocabulary size which increases its efficiency (Sánchez et al., 2013).

2.2.4.4 Vlad

Vlad descriptor is based on the aggregation of SIFT features into a low-dimensional feature vector (Yang et al., 2015). The construction of this vector is as follows (Arandjelovic and Zisserman, 2013):

- Extract SIFT features of size $128 - D$.
- Create k clusters and assign each descriptor to the closest cluster.
- For each cluster, sum the differences between descriptors and cluster centers.
- Concatenate the k $128 - D$ differences into a single k $128 - D$ vector.
- $L2$-normalize the final vector.

The main advantage of this vector is its compact size, which reduces the computational cost, however, it is not very robust when it comes to outliers (Husain and Bober, 2016).

2.2.5 Hashing

The idea of hashing is to transform high dimensional features into compact binary codes (Zhu et al., 2016). In a supervised learning method, the generated hash code should respect the semantic similarity between images by providing pairs of similar images (Wang et al., 2010). In such a case, the euclidean distance between images pairs should depend on the semantic similarity between images, where the higher the distance, the less similar images are (Xia et al., 2014).

A well known hashing algorithm for similarity search is Locality-Sensitive Hashing (Gionis et al., 1999), which was suggested to improve the search in high dimensional spaces such as large images repositories.

Examples of recent hashing algorithms are Liu et al. (2016), Yang et al. (2017), and Xie et al. (2017).

2.3 Implicit feature extraction techniques: deep convolutional features

Deep convolutional descriptors could be acquired using CNNs. Unlike linear models that are understandable, CNNs are non linear models that lack the interpretability, they are consequently considered as black boxes (Lipton, 2018). In other words, the inner functioning of these techniques is not transparent to humans, and it is not possible to understand how an output was generated (Silva et al., 2019). Despite of this, these features are becoming very popular in CBIR as they proved to be robust and led to accurate retrieval results (Hoang et al., 2017) (Laskar and Kannala, 2017) (Xu et al., 2018b) (Xu et al., 2018c) (Xu et al., 2018a) .

Images characterization using deep convolutional features is mainly divided into two directions. The first one consists in extracting off the shelf features from a pretrained CNN, without altering its architecture or its weights (Hoang et al., 2017) (Wei et al., 2017) (Xu et al., 2018a).

The second one consists in using a finetuned pretrained CNN to extract images features (Xu et al., 2018b) (Radenović et al., 2018) (Pang et al., 2018). Finetuning consists in training the last layers on a dataset from the same domain as the inference domain, such as landmark retrieval (Babenko et al., 2014) (Noh et al., 2017). The reason behind finetuning is to adapt the model to a given task, so it can extract more relevant features from the used dataset.

Generally, the model is cropped at the last convolutional or pooling layer (Laskar and Kannala, 2017) so as to consider all layers that perform feature extraction. When the convolutional layers are used, a pooling operation is generally carried out to aggregate these features (Gordo et al., 2017) (Xu et al., 2018c) (Xu et al., 2018b) . Moreover, these features are reduced using PCA algorithm (Wold et al., 1987), either in a preprocessing (Hoang et al., 2017) or a postprocessing step (Noh et al., 2017). The top removed layers consist in the fully connected layers that serve to classify an input to a given category, where the output is a class probability vector.

It must be underlined that CNN CBIR methods make use of existing pretrained CNNs (Simonyan and Zisserman, 2014) (Szegedy et al., 2015) (He et al., 2016a) (Chollet, 2017) (Howard et al., 2017) (Zoph et al., 2018) to extract the deep convolutional features, instead of creating and training a CNN from scratch. These CNNs architectures are considered as the state of the art in the imaging field, and they have seen enough patterns while being trained on 1 million images from ImageNet dataset.

2.4 Commonly used similarity and distance measures

After extracting images features in the offline phase as discussed above, it is possible to input a query to identify the most similar images in the database. In this step, the query is compared to all database images using either a similarity measure (Pang et al., 2018) or a distance function (Fachrurrozi et al., 2018).

As indicates its name, a similarity measure computes the degree of similarity between images, where the higher the similarity value, the more similar images are. On the other hand, a distance measure computes the distance between images using a distance function, where the higher the distance, the lower the similarity between images pairs. On top of these two measures, there exist other similarity matching measures that have been considered in the CBIR field, as in (Mosbah and Boucheham, 2017b), however, we give only some similarity and distance measures since these are the most commonly used in CBIR.

Considering I and J the N-dimensional feature vectors of the two images to compare, respectively, the distances and similarity measures are given in the following equations.

2.4.1 Manhattan distance (1-norm distance)

$$d(I, J) = \sum_{i=1}^{N} I_n - J_n \tag{2.22}$$

2.4.2 Euclidean distance (2-norm distance)

$$d(I, J) = \sqrt{\sum_{i=1}^{N} (I_n - J_n)^2} \tag{2.23}$$

2.4.3 Canberra similarity

$$s(I, J) = \sum_{i=1}^{N} \frac{|I_i - J_i|}{|I_i| + |J_i|} \tag{2.24}$$

2.4.4 Histogram intersection similarity

$$s(I, J) = 1 - \frac{\sum_{i=1}^{N} min(I[i], J[i])}{1 + I[i] + J[i]} \tag{2.25}$$

2.4.5 Cosine similarity

$$s(I, J) = \frac{I \times J}{||I|| \times ||J||} \tag{2.26}$$

2.4.6 Dot product similarity

$$s(I, J) = ||I|| \times ||J|| \times \cos\theta \tag{2.27}$$

2.5 Performance evaluation metrics

In this section, we describe the metrics to evaluate the goodness of CBIR systems. The computation of these metrics is based on some measures, which are given below.

True Positives (TP) Retrieved and relevant elements.

True Negatives (TN) Not retrieved and irrelevant elements.

False Positives (FP) Retrieved and irrelevant elements.

False Negatives (FN) Not retrieved and relevant elements.

It should be highlighted that in CBIR systems, we aim to maximize the TP and TN measures and minimize the TN and FN measures so as to increase the retrieval accuracy.

Performance evaluation metrics are discussed below.

2.5.1 Precision (P)

This metric computes the ratio of the number of relevant images retrieved with respect to the total number of images retrieved. The precision suffers from several drawbacks: Since only the true positives are considered, this metric does not reflect any information about the relevance of the elements that are not returned, nor report the goodness of the model in terms of ignoring negative samples (the true negatives) (Powers, 2020). In addition, it doesn't take into account the results ranking (Raghavan et al., 1989), for instance, Given a set of returned elements at a given threshold, ranking the true positives higher or lower than the false positives will result in the same precision value.

The precision formula is given as follows:

$$P = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ retrieved\ images} \tag{2.28}$$

Which is equivalent to:

$$P = \frac{TP}{TP + FP} \tag{2.29}$$

2.5.2 Recall (R)

This metric computes the ratio of the number of relevant images retrieved with respect to the total number of relevant images. The recall suffers from the same drawbacks of the precision metric.

The recall formula is given as follows:

$$R = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ relevant\ images} \tag{2.30}$$

Which is equivalent to:

$$R = \frac{TP}{TP + FN} \tag{2.31}$$

2.5.3 Mean average precision (mAP)

This measure takes into account both precision and recall. It reflects not only the ability of a system to retrieve relevant elements, but also the ranking of these elements by computing the precision and recall at different thresholds. Consequently, this measure is well suited to compare between systems, since it gives a better evaluation of the performance by measuring the precision at every recall (Raghavan et al., 1989)

Similarly to precision and recall, this measure evaluates models based on binary judgments: whether an element is relevant or not. Whereas, correctly retrieved documents may have different degrees of relevance to the query, which is not considered by this measure and is regarded as a problematic point when assessing the goodness of retrieval systems (Kishida, 2005). This metric is computed as follows:

$$mAP = \frac{1}{L}\sum_{l=1}^{L} AP_l \tag{2.32}$$

Where AP_l is the average precision for a query l, and L is the total number of queries.

The average precision is defined by:

$$AP_l = \sum_{k=1}^{K} P_l@k \times (R_l@k - R_l@(k-1)) \tag{2.33}$$

Where K is the number of the required iterations to achieve a recall equals to 1. The lower k, the higher the mAP is: The lower the number of iterations k required to achieve a perfect recall, the higher the ranking of relevant elements.

2.5.4 F-measure

F-measure combines both precision and recall by their harmonic mean to give the best balance between them (Perez and Olague, 2008). This metric has the same limitations of precision and recall, regarding the binary judgments and the ignorance of true negatives. It is defined as follows:

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.34}$$

2.6 Location-dependent queries

After measuring the similarity between images, one can extract the similar images using many different techniques, called "location-dependent queries". In our thesis, we consider two types of location-dependent queries: KNN search and within-distance-query. These two techniques are discussed in the two following subsections.

2.6.1 K Nearest Neighbor (KNN) search

After ranking database images based on their similarity to the query, the KNNs could then be identified, so that:

$$d(Q, N_i) <= d(Q, N_{i+1}), \tag{2.35}$$

or

$$s(Q, N_i) >= s(Q, N_{i+1}) \, where \; i = 1, k-1 \tag{2.36}$$

Where d and s represent the distance function and the similarity measure, respectively, Q is the query image, i is the neighbour rank, and N_i is the i-th nearest image.

This technique allows the user to control the number of images to retrieve. However, since images repositories are constantly updated by new images, the user ignores how many similar images exist in the database, given a query. This has the effect that the number of images k to retrieve may be lower or higher than the existing relevant images. In the first situation (where k is lower), there is a lack of completion in the results since more relevant images exist, though, they are not retrieved. In other words, the number of false negatives is increased.

False negatives and true positives are negatively correlated, i.e. when the number of false negatives increases, the number of true positives decreases, since a relevant sample can be either a false negative or a true positive, which decreases the recall evaluation metric (see equation 2.31). On the other hand, the chances of retrieving a true positive at lower k values are generally better. In other words, the number of correctly retrieved images gets closer to the total number of retrieved images k, when k is reduced, which enhances the precision metric (see equation 2.28).

In the second situation (where k is higher), retrieving more images than the existing relevant ones may lead to an increase in the number of false positives, especially if they are ranked lower than the true positives, which decreases the precision metric (see equation 2.29). However, since this increases the chances of retrieving more relevant images (The true positives), the recall is improved (see equation 2.31).

Both situations reflect the well known precision-recall trade-off.

2.6.2 Within-distance-query

Within-distance-query is a particular type of range query where the range is circle (Sato and Narita, 2013). According to the same paper, range query may include other location dependant query types depending on the range, such as window-query, when the range is rectangular. Within-distance-query extracts images within a given distance/similarity range with the query

image, so that: $d(Q, I) < \tau$ or $s(Q, I) > \tau$ where d and s represent the distance function and the similarity measure, respectively, Q is the query image, I is a retrieved image, and τ is the used threshold value.

One way to define the threshold is by experimenting with several values and selecting the value that maximizes either the precision or the recall (Hamreras et al., 2019), or leads to a good compromise between the two metrics (Hamreras et al., 2020). Alternatively, it is possible to define an automatic way to set the threshold as in Yang et al. (2018).

Similarly to the KNN technique, it is difficult get rid of the precision recall trade-off, since in most cases retrieved images are not ranked correctly. As a result, the defined range may include irrelevant samples and relevant samples may be outside this range. In such a case, the source of error is back to indexation and similarity measurement phases, if we consider that the threshold is correctly defined.

Based on this, enlarging the range would result in the retrieval of more positive and negative samples, which increases the recall and decreases the precision. However, reducing this range would result in the opposite situation, as explained for the KNN retrieval type.

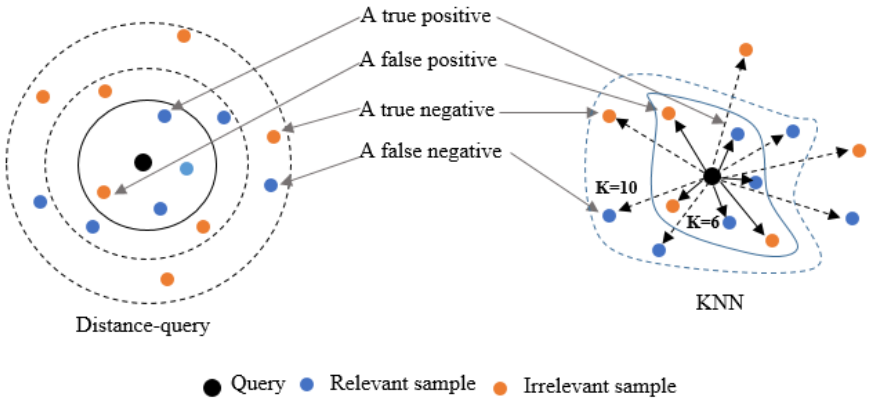


FIGURE 2.7: range query and KNN techniques. Dashed circles and arrows indicate other larger values for the range and the number of neighbours k.

For both retrieval types, our assumptions are justified by the results obtained in Chapter 6 and Chapter 7.

2.7 Benchmark datasets

There exists several images datasets to assess the retrieval performance of CBIR systems. Below, we describe some of them.

Oxford5k (Philbin et al., 2007) [5] Includes 5062 images collected from Flickr [6]. Images are organized in 11 categories, that correspond to Oxford landmarks. 5 queries are given per class, in total 55 queries. In addition, ground-truth images are given, where images could be "good", "ok", "bad", or "junk", depending on their relevance to each query, where "good" and "ok" are considered as relevant.

Paris6k (Philbin et al., 2008) [7] Compromises 6412 images collected from Flickr. Similarily to Oxford5k, images are organized in 11 categories depicting Paris Landmarks, and each category is represented by 5 queries. Moreover, images could be "good", "ok", "bad", or "junk", where "good" and "ok" are relevant to the query.

Oxford105k and Paris106k (Philbin et al., 2007) Are the results of combining 100,071 distractor images from Flickr dataset with Oxford5k and Paris6k, respectively. This combination allows to evaluate the retrieval performance at a very large scale.

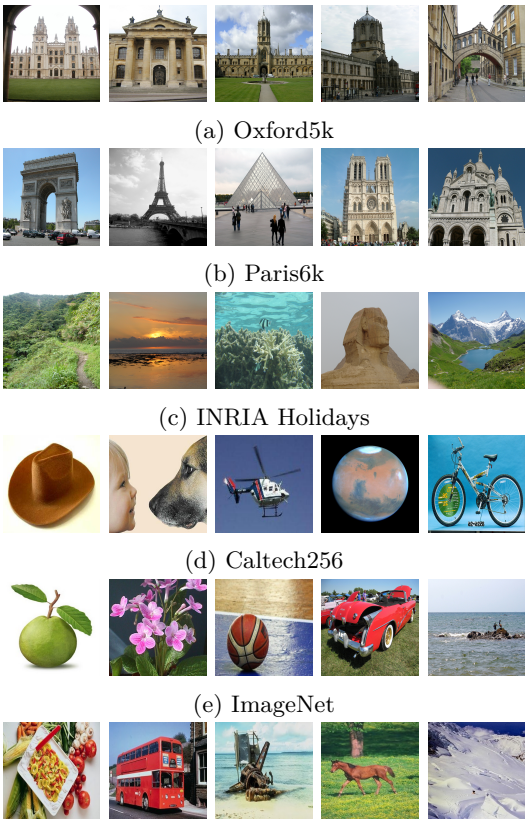
Holidays [8]**(Jegou et al., 2008)** Includes 1,491 images, from which 500 are queries. All these represent 500 scenes of personal images holidays.

Caltech256 [9]**(Griffin et al., 2007)** Contains 30,608 images, grouped into 256 semantic categories. Moreover, it exhibits at least 80 images per category.

ImageNet [10]**(Russakovsky et al., 2015)** Basically, this dataset is used for classification tasks, especially in the evaluation of Convolutional neural networks, however, it has been also used in CBIR. ImageNet includes 1.2 million images for training, 50,000 images for validation, and 150,000 images for testing, all grouped into 1000 categories.

Corel [11] Depending on its size, this dataset is referred to as Corel10k, Corel5k, and Corel1k, which are subsets from this dataset with 10,000, 5,000 and 1,000 images, equally grouped into 100, 50, and 10 categories, respectively.

[5] `https://www.robots.ox.ac.uk/~vgg/data/oxbuildings/`
[6] `https://www.flickr.com/`
[7] `https://www.robots.ox.ac.uk/~vgg/data/parisbuildings/`
[8] `http://www.lear.inrialpes.fr/people/jegou/data.php/`
[9] `https://www.kaggle.com/jessicali9530/caltech256/`
[10] `http://image-net.org/challenges/LSVRC/2010/index/`
[11] `http://wang.ist.psu.edu/docs/related/`



(a) Oxford5k

(b) Paris6k

(c) INRIA Holidays

(d) Caltech256

(e) ImageNet

(f) Corel

FIGURE 2.8: Some CBIR benchmark datasets samples.

2.8 Conclusion

In this chapter we made an overview of the CBIR area including the related concepts, the used techniques, and performance evaluation metrics. The presented techniques range from feature extraction techniques, distances and similarity measures to location dependent queries.

Speaking of advanced CBIR systems, the overviewed feature extraction techniques are not to apply directly unless they are selected, tuned, or combined in some way to boost their performance. In one word they should be “adapted” using an adaptive technique.

Instead of jumping directly into the used adaptive techniques in the CBIR field, the next chapter will spot the light on the adaptation paradigm to clarify its concept as well as the major adaptive techniques used in the literature.

Chapter 3

Dominant adaptive techniques in the literature

This chapter highlights the constant need of adaptation in a lot of areas, including computational systems. We discuss the most ubiquitous adaptive techniques and some of their recent applications in order to show how the adaptation is improving several areas.

3.1 A general overview of adaptation techniques

Adaptation was inspired by biology, where an adaptive system is defined by "An advantageous conformation of an organism to changes in its environment" (Narendra and Annaswamy, 2012). In a general context, the adaptation aims at changing a system's behaviour depending on some criterion in order to get a better response in terms of effectiveness and/or efficiency. In fact, the adaptation is not limited to computational systems, it has influenced many areas in order to enhance them by overcoming some serious challenges. Below, we describe how the adaptation paradigm is influencing some non computational areas.

Biological immune system First of all, adaptation is a natural phenomena that could be found in the biological human system, which - instead of developing antibodies for all possible antigens - it adapts the antibodies to antigens whenever they attack the body (Holland, 1992a).

Education Adaptive learning is one of the preoccupations in education that may significantly boost the learning experience. One of its aspects is courses customization depending on the student profile, which has been mainly addressed through the learning styles (Chen and Zhang, 2008) (Chang et al., 2009) (Tortorella and Graf, 2017) (Díaz et al., 2018) (Li, 2019), where the material presented to the student is adapted based on his learning style.

Industry One of its aspects is the adaptation of firms to the new industry revolution, known as Industry 4.0 or the fourth generation industry. This latter is characterized by the increase in digitalization and automation of several industrial manufacturing processes using a state of the art IT techniques, so as to increase the efficiency production and reduce the required cost (Hamada, 2019). According to the same paper, firms are using several IT techniques to achieve this, as examples: Cloud computing and real time big data analytics may lead to a decrease in the production cost, predictive maintenance to avoid any system crash, as well as discovering possible optimization strategies. Other possible adaptation strategies are networking between manufacturing sites, artificial intelligence and internet of things (Hamada, 2019).

Agriculture In the agriculture area, the adaptation is carried out with respect to the climate change. Indeed, the change in climate may greatly affect the agricultural crop, consequently, it is necessary to take this into consideration to enhance the quantity and the quality of the plantings. In this regard, two main techniques are approved in traditional agriculture, namely raised fields, which are used to avoid some natural disasters, such as floods, and Dryland farming, which is used in agricultural areas during dry seasons to cover the lack of rainfall (Altieri and Nicholls, 2017).

Tourism As an example, the ski tourism is vulnerable due to the climate warming. One of the most common solutions is to adapt to the climate change by snowmaking, which allows to produce artificial snow (Steiger and Scott, 2020).

3.2 Dominant computational intelligence techniques as adaptation tools

In the computation area, adaptation aims to tune systems parameters given a problem instance, so as to increase their outcome (Narendra and Annaswamy, 2012). Since learning is the process of parameters optimization based on training data or past experiences (Singh et al., 2017), the

adaptation could be injected into computing systems by training them, meaning that every trained system is adaptive. As such, the techniques presented in the remainder of this chapter are said to be "adaptive" for the reason that they need to be trained in order to properly perform the assigned task.

3.2.1 Machine learning (ML) based techniques

ML is a subfield of artificial intelligence, where algorithms are able to learn and adapt their outcome for a given task without being explicitly programmed (Simon, 2013). Instead, the learning in this case is based on the training data or past experiences (Alpaydin, 2020). There exists several machine learning algorithms that have been successfully applied in a wide range of areas. In this subsection, we describe some of them.

3.2.1.1 Artificial Neural Networks (ANNs)

ANNs are inspired by the brain functioning. The architecture, as well as the functioning are depicted in Figure 3.1 .

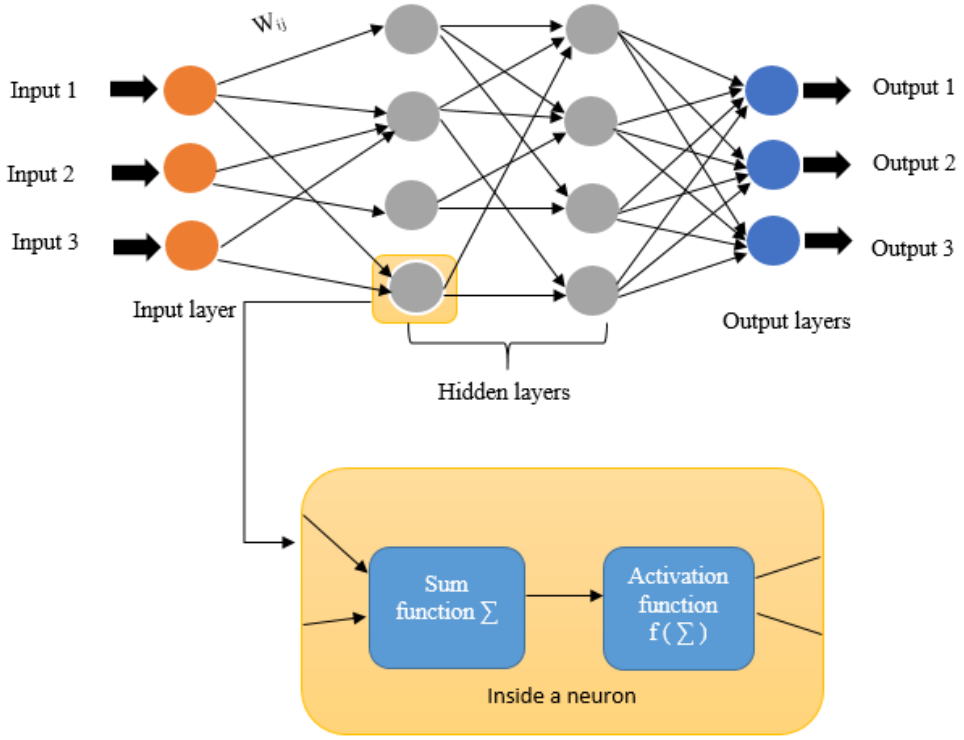


FIGURE 3.1: The ANN architecture including the different layers and the neuron functioning.

As shown in the figure, the neurons are organized in stacked layers, and the number of these layers varies from two layers (input and output layers) to several layers (ANN with hidden layers). The neurons of each layer are connected to transfer information to the subsequent layer. These connections are weighted in a manner to emphasize or de-emphasize the information output from a neuron depending on the assigned weight. Moreover, each neuron has an activation function to compute the output and transfer it to the next layer. After a neuron receives the outputs of the connected neurons from the previous layer, these outputs are aggregated and passed to an activation function, the role of which is to activate or deactivate this neuron. The aggregation function is given in the following equation:

$$C_i = \sum_{j=1}^{J} w_{ij} x_j \tag{3.1}$$

Where C_i is the output of the i-th neuron, J is the number of connected neurons from the previous layer, w_{ij} is the weight of the connection between the i-th neuron and the previous j-th neuron, and x_j is the output of the j-th neuron.

There exists several activation functions used by ANNs. RELU is the most common used activation function (Albawi et al., 2017) which acts only on negative inputs to transform them to a null value. This function is given in the following equation:

$$RELU(x) = max(0, x) \tag{3.2}$$

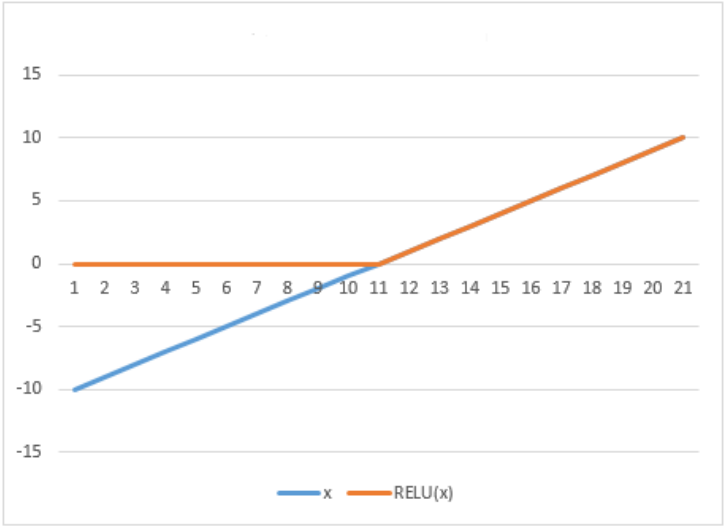


FIGURE 3.2: RELU activation function curve.

Additionally, ANN layers can be fully connected, i.e. each neuron from a given layer is connected to all neurons in the subsequent layer, or partially connected if at least one neuron is not connected to another in the following layer.

Based on their architecture, ANNs can be classified into two categories (Jain et al., 1996):

- **Feed-forward ANNs** They have only forward connections with no loops. An example of this are convolutional neural networks (CNNs) that take images as input and classify them into categories.
- **Recurrent ANNs** In addition to forward connections, the feedback connections allow to form loops. After computing the output of a given a neuron, if one of its connected neurons in the subsequent layers has a feedback connection to that neuron, then the output is passed as a new input to that neuron, and the output is computed again. This way the ANN provides several outputs or a sequence of values. Consequently, this type is used in applications that deal with sequenced information to maintain dependency between data sequences. An example of this are LSTMs that are used for speech recognition (Han et al., 2017) and natural language processing (Chen et al., 2016). In addition to being recurrent, LSTMs they are characterized by memory cells, where input, output, and forget gates are used to modulate the information flow (Yao et al., 2015).

After building the right architecture, the ANN is trained on a relevant dataset to update its weights and make it a productive model. One of the most common learning algorithms is back-propagation. This algorithm is used for pattern classification based on supervised learning (Jain

et al., 1996). The latter is described below:

Algorithm 1: Back-propagation learning algorithm as explained by (Jain et al., 1996).

1.Initialize the weights to small random values;

2.Randomly choose an input pattern x^{μ};

3.Propagate the signal forward through the network;

4.Compute γ_i^L in the output layer ($o_i = y_i^L$);

$\gamma_i^L = g'(h_i^L)(du_i - y_i^L)$‘

Where h_i^l represents the net input to the i-th unit in the l-th layer, and $g\prime$ is the derivative of the activation function g.

5.Compute the deltas for the preceding layers by propagating the errors backwards;

for $L = L - 1, .., 1$ **do**

$\gamma_i^l = g'(h_i^l) \sum_{w_{ij}^{l+1}} \gamma_j^{l+1}$

end

6.Update weights using:

$\triangle w_{ij}^l = \eta \gamma_i^l y_j^{l-1}$

7.Go to step 2 and repeat for the next pattern until the error in the next layer is below a prespecified threshold or a maximum number of iterations reached.

ANNs have been applied to many tasks. An example in information security is Yin et al. (2017) that uses a recurrent neural network for intrusion detection. The model was trained and evaluated with different parameter settings, including the number of neurons in hidden layers, as well as the learning rate. Moreover, it was trained in two ways: binary classification to detect a normal/abnormal behavior, and multi classification to distinguish between different intrusion types. The use of a RNN led to a high classification accuracy, and outperformed other evaluated machine learning based techniques. Another example of the feed forward ANN which was used for medical diagnosis, particularly lung cancer detection is Nasser and Abu-Naser (2019). the used ANN is a DNN with one hidden layer. Moreover, it has 14 inputs depicting disease symptoms such as allergy, shortness of breath...etc. The latter was trained on a relevant dataset for binary classification: presence or absence of cancer, and achieved a high accuracy of 96.67%.

3.2.1.2 Support Vector Machines (SVMs)

A SVM classifies data into two classes, that is binary classification. A training phase is required to train the hyper-plane that separates data in a linear or non linear way (Shen et al., 2016). The non linear separation is achieved through kernel functions, which allows to make non linear decision boundaries for linear classifiers (Ben-Hur and Weston, 2010). In the training phase, the algorithm attempts to maximize the margin between the two categories (Lin and Wang, 2002), and thus provides a more accurate classification, as shown in Figure 3.3.

It should be considered that more recently, SVMs are used for multiclassification tasks. A common strategy to achieve this is "one-versus-all" classification, where multiple binary classifiers are used, and the hyper-plane separation occurs between one class and the rest of classes; the belonging class for an instance is the one with the maximum margin (Schütze et al., 2009).

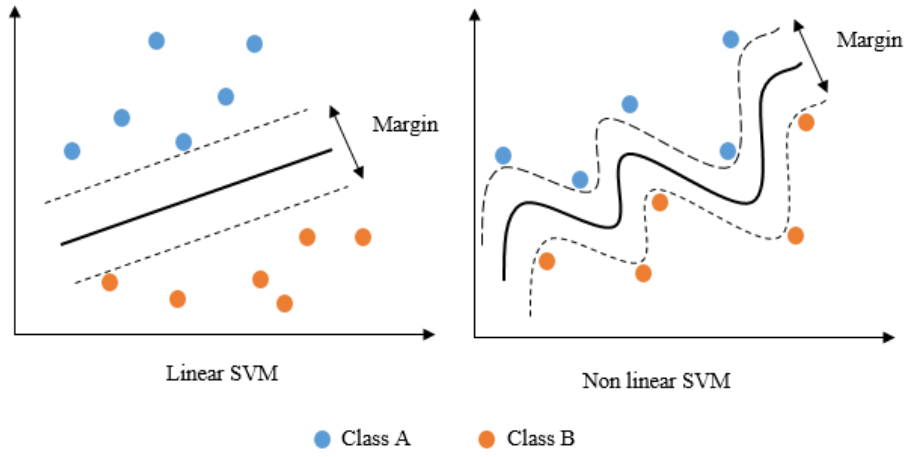


FIGURE 3.3: The two types of SVMs classifiers: linear and non linear.

SVMs are widely used in the literature. For example, Huang et al. (2017) uses SVMs and SVMs ensembles for breast cancer prediction. Authors evaluate SVMs peformance by training them with different kernel functions. Moreover, they built an ensemble of SVMs to improve their outcome using the bagging and boosting techniques. The evaluation of these parameters with different combinations led to the selection of the SVM kernel and the ensemble technique depending on the dataset size, which led to a high classification accuracy. An earlier, but a quite interesting application of SVMs is spam categorization (Drucker et al., 1999). This paper evaluate SVMs in two different manners, either by selecting the most relevant features from the dataset or using all the dataset which consists of a set of messages (spam and non spam). In this case, features are words and the message is converted to a feature vector containing various words, so that the SVM is used to classify the feature vector into spam or non spam message.

A comparison with a Ripper, Roccio, and boosting decision trees showed that SVMs are better performing in terms of training time, and have acceptable error rate compared to the same algorithms.

3.2.1.3 Decision Trees (DTs)

A DT is a simple machine learning algorithm used to classify data through a successive feature analysis, i.e. data features are depicted by nodes, where in each step features are analyzed to decide the next state (feature), until the class is identified in the last node. This algorithm is composed of several nodes, where each node has children nodes (except leaves) and each parent node is connected to its children. The connection to each children node is based on the satisfaction of a condition which allows to move to a new state. The final nodes indicate the possible classes of input data.

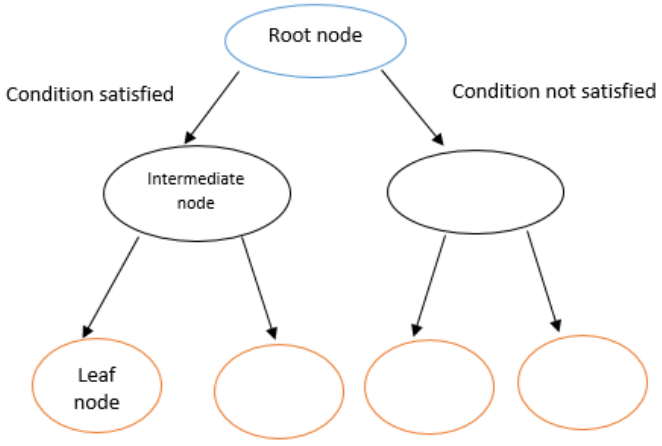


FIGURE 3.4: A decision tree architecture with three levels.

DTs are commonly used in the literature. As an example, Phu et al. (2017) used a decision tree algorithm (ID3) for english document emotional classification. The algorithm was trained on a wide dataset containing 115,000 english sentences, and tested on 25,000 english document. Documents are first split into sentences, the classification is then based on the number of positive/negative rules in each sentence, where a sentence having more positive rules is classified as positive, otherwise, it is classified as negative. This method led to an acceptable classification rate of 63.6%. Another example is theft detection in smart grid (Jindal et al., 2016). This paper uses a decision tree to predict the electricity consumption for the consumers based on several inputs such as season and temperature. After that, the output of the decision tree, which is the

electricity consumption is input to a SVM to classify electricity usage into normal or fraudulent. Using this method, a classification accuracy of 92.50% was reached.

3.2.1.4 Clustering

Clustering is an unsupervised learning algorithm. The concept is to create clusters of data, where data belonging to the same cluster share common characteristics, and thus belong to the same category.

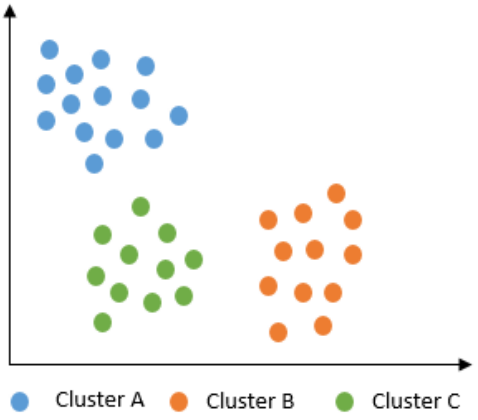


FIGURE 3.5: An example of three data clusters resulting from applying a clustering technique.

One of the most common clustering algorithms is k-Means, which is given below.

Algorithm 2: k-Means algorithm.

1. Randomly choose k data points as clusters centers;
2. Assign each data to the nearest cluster center;
3. Recompute the clusters centers based on included data;
4. If a convergence criterion is not met, go to step 2, otherwise, stop the algorithm.

An application example of clustering is Fan et al. (2018). In this paper, the clustering is employed with a CNN for person re-identification. First, a CNN is trained on a dataset from the same domain (persons with different camera positions) to initialize it with appropriate weights. After that, deep features are extracted from an unlabeled dataset and grouped into clusters, where each cluster designs a person. These clusters are then enhanced by keeping only the nearest features to centroids. Features extraction and clusters enhancement are carried out in a progressive way, until the clusters are representative enough. Finally, finetuning allows to adapt the original CNN to the unlabeled dataset so as to improve its classification power.

In Chang et al. (2017), the authors propose a novel technique for pair-wise image classification, which is "deep adaptive clustering" (DAC). This technique deals with the issue that input images to a cluster are unlabeled, and thus the similarity between them is unknown. To tackle this issue, they propose to train a CNN on unlabeled images dataset to generate features that consist in one hot vectors, so that a cosine measure is applied to measure similarity between them and decide whether pair images are similar or dissimilar. After that, pair images are selected based on their similarity as inputs to train the CNN. The whole process from feature generation to CNN training is carried out again, until all the unlabeled images are considered for training. Finally, similar samples are included in the same cluster.

3.2.2 Evolutionary computing (EC) and swarm intelligence (SI) techniques

EC and SI are another subfield of artificial intelligence which is widely used in optimization problems (Zhang et al., 2011). It consists of algorithms that are inspired by natural evolution to solve these problems. In this subsection, we present some well known evolutionary computing techniques as well as some of their applications.

3.2.2.1 Genetic Algorithms (GAs)

The concept of GAs is to create a descending population of solutions based on natural selection, which consists in crossover, mutation, and inversion operations (Mitchell, 1998) to the extent of improving the population in each iteration of the algorithm. The motivation behind the development of GAs was to inject the natural adaptation phenomena into informatics systems (Mitchell, 1998). GAs are used in solving optimization problems where the search space is so huge (Holland, 1992b), that a sequential search technique will fail to solve it in a polynomial time. In GAs, each individual in the population is depicted by a chromosome. The natural selection then allows to solve those problems by selecting partial solutions from these chromosomes (genes) to create generations that are more converged toward the desired output (Holland, 1992b).

A recent application of GAs to face recognition is Zhi and Liu (2019). First, PCA is used for gray features extraction and reduction. After that, a GA is used for features search, where each chromosome is decoded as an 8-bit binary code and used to represent a segment of an image having a length of 8 bits. Genetic operations such as crossover and mutation are used to create

a better population in each iteration of the algorithm. In each iteration, the chromosomes with highest fitness values are saved, so that the chromosome having the best value is selected as a segmentation value at the end of the algorithm. Finally, images classification is carried out using a SVM.

Hosseinabadi et al. (2019) used GAs for solving open-shop scheduling problem (OSSP). In this paper, jobs are represented by chromosomes, where each gene refers to an operation of this job, so that a chromosome represents a schedule for running operations in a specific job. Moreover, parents are selected as chromosomes having best fitness from a subset of randomly chosen chromosomes, and new individuals are acquired thanks to cross-over and mutation operations. The next generation, i.e. jobs with new operation scheduling is produced with the 10% best chromosomes in terms of their fitness values. Finally, the best chromosome is chosen after running the algorithm for a defined number of iterations.

3.2.2.2 Ant Colony Optimization (ACO)

ACO is inspired by the ant behaviour to find the shortest path to their food. The route is composed of different paths, where, at each node a path is chosen to reduce the distance to the final destination. The usefulness of a given path is indicated by the pheromone quantity left by each ant after passing through it. That way, each ant contributes to the solution by updating the pheromone quantity (Eren et al., 2017), so that the shortest route will be reinforced by the pheromone and will be followed by most of the ant colony individuals (Saka et al., 2013). In order to achieve this, all possible paths are depicted by a graph, where a path is a unidirectional arc, weighted based on its usefulness (the pheromone quantity) (Deb, 2011).

In each iteration of the algorithm, the weights are updated following the equation below (Saka et al., 2013).

$$\tau_{i,j}(t+n) = \rho\tau_{i,j}(t) + \Delta\tau_{i,j}(t) \tag{3.3}$$

Where $0 \leq \rho < 1$, $1-\rho$ is the evaporation amount of the pheromone between two instants t and $t+n$, and $\Delta\tau_{i,j}$ is the difference in pheromone quantity between two points i and j.

The process is done when all ants have passed through the route and updated the pheromone quantity (Eren et al., 2017). Based on the latter, the best route could then be identified.

One of the most interesting applications of ACO is feature selection. The feature selection process is transformed into an ACO problem by considering features as nodes, where the paths lead

to another feature (Uthayakumar et al., 2020). In this work, which introduces an ACO-based feature selection and classification algorithm for financial cresis prediction. Before updating the paths weights, selected features are first evaluated, they are then updated based on their relevance in order to select a minimum subset of features having a high representational power for the problem at hand.

Another example is road sign detection and recognition (Jayaprakash and Keziselvavijila, 2019). In this paper, color features are extracted and reduced to a subset using ACO algorithm. The classification is finally carried out using an ensemble of SVM classifiers using a majority voting.

3.2.2.3 Particle Swarm Optimization (PSO)

PSO was developed by Kennedy and Eberhart (1995). It is inspired by the animal's social behaviour such as birds and ants (Yang and Karamanoglu, 2013). In this algorithm, a particle consists in an individual that aims to find a better solution based on its own experience as well as the experience of its neighbours, resulting in a local and a global search processes (Sajja and Akerkar, 2013). This way, in each step, individuals collaborate to converge toward a better solution in the search space to finally find out a quasi-optimal solution.

PSO has been widely used in the literature. Hossain et al. (2019) applied PSO to university course scheduling problem. In this case, a particle is the full schedule for instructors, students, classrooms and laboratories, which is divided into M parts, representing each the schedules for each instructor. The algorithm first initializes all particles randomly, then calculates the best solution which is considered as the global best solution. The update of a particle's solution in each iteration is based on swapping courses for the same instructor with respect the the previous best solution and the global solution. Finally, the final solution is found using a selective search algorithm that takes into account all intermediate solutions. The intermediate solution having the highest fitness value is selected.

Another application in medicine is (Kavitha and Prabakaran, 2019). The latter uses PSO and fuzzy C-means clustering to enhance the detection of lung cancer. First, the input image is denoised and segmented. After that, a PSO is used to find the most accurate value of the clustering centers of fuzzy C-means clustering. Herein, the particles consist in clusters that are updated in each iteration, and the PSO enhances the clustering algorithm by considering neighborhood information.

3.2.3 Multi-agent systems (MASs)

A MAS is composed of agents that collaborate to solve a problem which is likely to be unsolved by an individual agent (Flores-Mendez, 1999). Even though each agent has its own view for the problem and acts in a complete independent manner, interaction with other agents is needed to solve the targeted problem (Huberman and Clearwater, 1995). An agent is an entity, let's say a process that is mainly characterized by (Flores-Mendez, 1999):

Adaptivity which is the ability to adapt its behaviour to its environment.

Autonomy which means that its behaviour is self-made and self-controlled.

Collaborative behaviour which means the ability to collaborate with other agents to solve a problem.

MAS systems have been successfully applied to many problems. An example of MAS applications is resource allocation in distributed systems (Banerjee and Hecker, 2017). In this paper, each computer is depicted by an agent, allowing to create a decentralized system. Moreover, each agent communicates with its neighbours in order to dynamically create a cluster to execute a queue of processes in parallel.

Another example is González-Briones et al. (2018). This paper uses a MAS for the classification of gender and age from images, where agents where organized in different layers depending on the assigned task. These tasks vary from images acquisition, preprocessing, training, and classification. Moreover, a workflow agent organizes the succession of these operations by communicating with these agents.

3.2.4 Fuzzy Logic (FL)

The FL is based on the idea that everything is a matter of degree (Kosko and Toms, 1993). According to Zadeh (1988), fuzzy logic is about finding an approximate answer in an environment where the provided information is incomplete, non-reliable, or inexact. A good example to explain fuzzy logic given by Kosko and Toms (1993) is the following: If we consider only one set which is zero, then the zero belongs to this set by 100%, and all other numbers do not belong to it (all or none). The fuzzy logic allows to replace the all or none membership by degrees of memberships, i.e. all other close numbers may belong as well to this class by different ratios

depending on their closeness to the number zero, such as 80%, 50%. As such, when we can not determine the all or none membership of an entity to a given subclass, then we can approximate its membership to this class with a degree of certainty.

Ramos et al. (2019) uses a fuzzy logic technique for fault diagnosis in industrial systems. The proposed technique first computes residual vectors which consist in the difference in the system output in normal/abnormal state, where two classes: zero and one, represent these states, respectively. Afterwards, the trapezoidal membership functions are defined to evaluate these residual vectors. The latter are finally compared to a threshold value to determine whether they reflect an abnormal behaviour or not.

Another interesting application of fuzzy logic is speech recognition (Eljawad et al., 2019). After extracting the features using DWT, a neural network is used to train the fuzzy system's rules and to build the internal structure of membership functions. The input of the system is the extracted features and the output is the speech segments ID's. Finally, the trained fuzzy system can be deployed to recognize the speech.

3.2.5 Rule Based Systems (RBS) or Expert Systems (ES)

RBSs mimic the human reasoning for solving complex tasks (Grosan and Abraham, 2011) . Herein, the knowledge is organized in conditional statements in a form of if-then rules, easily understood by humans (Kovarik Jr, 2006). This knowledge is used by an inference engine that can deduce new knowledge based on the input by applying several operations such as deduction, abduction and transitivity (Laurini and Thompson, 1992). Moreover, the selection of adequate rules and adapting their sequences to a given task allows to solve complex problems (Hayes-Roth, 1985). An expert system is characterised by being (Buchanan and Duda, 1983):

Heuristic Which means the ability to solve problems based on judgment and formal knowledge, which allows to solve non polynomial problems.

Transparent Which means that it clearly shows how a given problem has been solved.

Flexible Which gives the possibility to add new rules.

An example application of this technique is Kolekar et al. (2019). It uses a RBS to integrate adaptation in E-learning systems by creating an adaptive user interface. The rules show/hide

the learning objects based on the student learning style. This way, if a student learning style is visual, visual components are shown, if it is sensing, more examples and illustrations are suggested... and so on.

Another application is project risk assessment (Yang et al., 2019). First, a random sampling is applied to the rule base to create several subspaces. After that, a belief rule based (BRB) could be generated based on each subspace. Each BRB is based on belief rules, where the attributes are risk weighted factors and the input is the project performance with a belief degree. Rules and attributes weights, as well as belief degrees are optimized during a learning process. After that, an ensemble of independent BRB from different subspaces is created to optimize the performance.

3.3 Conclusion

To sum up, the adaptation paradigm has established itself in many fields and let to satisfactory outcomes. In this chapter, we aimed to prove this by describing well-known adaptive techniques and showing how they enhanced computational systems. In the next chapter, we go straight to the point and present the adaptation as a tool to empower CBIR systems by underlying the main adaptive techniques within this area.

Chapter 4

Adaptation as a tool for empowering CBIR systems

We suggest to improve CBIR systems by applying the most adequate technique depending on the problem instance, so as to increase the performance of these systems on the problem meant to be solved. From our own perspective, this could be achieved by the use of an adaptive technique, whether by selecting the most convenient feature(s), distance, relevance feedback technique, or parameter(s), depicted in the selection paradigm, performing a post-processing technique to enhance the initially retrieved elements, using information retrieval tools, or learning the algorithm parameters using a machine learning technique. This chapter is dedicated to the study of the omnipresent adaptive techniques in CBIR.

4.1 Selection based techniques

The selection paradigm aims to select the most convenient feature(s), distance, relevance feedback technique, or parameter(s) such as the threshold, depending on the target CBIR task, resulting in four main selection techniques, which are feature selection, distance selection, relevance feedback technique selection, and threshold selection. In this section we explain the concept of each of them and give some of their CBIR applications.

4.1.1 The essence of the selection paradigm: the No Free Lunch Theorem (NFLT)

The NFLT is the basis for the selection paradigm. Ho and Pepyne (2002) defines the NFLT as an impossibility theorem: " It is impossible to develop an optimization strategy that is universally better than all others", simple but deep as a definition. By "universally" they mean all problem instances meant to be solved. Thus, if a strategy is judged as the best performing one, then this judgment is true only given a specific subset of problem instances.

Two main facts are related to the impossibility theorem (Ho and Pepyne, 2002): First, all optimization strategies have the same average performance measured over all possible problem instances. This means: If a strategy outperforms another on a subset of problems, then it must be outperformed by the same strategy on another subset of problems. Second, if a strategy have more than the average performance given a problem subset, then it must have less than the average performance given another problem subset. This second fact is known as "The conservation of performance law" and is the essence of the NFLT.

4.1.2 Feature Selection (FS)

The variety of information included in images gave way to the existence of several descriptors, called features. Although these features offer rich and diverse representations for images, using all candidate specific domain features for a target task is not common. The reason is that these features may be irrelevant to this task, or they are redundant in the sense where they do not provide any new information about the target task (Dash and Liu, 1997). Redundancy could also manifest in highly correlated features, where one of the highly correlated features is sufficient to represent data (Chandrashekar and Sahin, 2014).

In the light of these considerations, there is a need to *select* only the features that better suit the target task. FS approaches not only lead to improve the system's accuracy by improving data representation, but also help to reduce the computational cost by reducing the set of features, and better understanding data (Dash and Liu, 1997) by exploring it.

According to Motoda and Liu (2002), feature selection could be part of a data mining process, as the aim of data mining algorithms is to discover relevant patterns in data. Additionally, it could be carried out separately as a prepossessing step to prepare data for mining. Indeed,

feature selection is one of the omnipresent prepossessing techniques for data mining (Da Silva et al., 2011). This second approach is more interesting and offers several benefits (Motoda and Liu, 2002): First, it can be done once and used for several data mining tasks. Second, it is less computationally expensive compared to data mining algorithms, and last, several feature selection algorithms could be evaluated since this step is done offline.

It is important to highlight that feature selection process is other than feature extraction process. Both methods are considered as dimensionality reduction approaches (Da Silva et al., 2011) and aim to increase the representational power of features. However, feature extraction proceeds by generating new features based on originally extracted ones (Khalid et al., 2014). Whereas, feature selection reduces the set of features into a relevant subset.

Feature selection process follows four major steps (Dash and Liu, 1997):

Generation process Where subsets of features are generated for evaluation. Three main search types could be carried out: forward sequential, if the set is initially empty and features are added gradually, backward sequential if the set is initially full and features are gradually removed, or random, where all possible subsets have approximately equal chances for being generated (Motoda and Liu, 2002).

Evaluation Each generated subset of features is evaluated using an evaluation function such as Error and Divergence (Khalid et al., 2014).

Stopping When a stopping criterion is met (a predefined number of iterations or features), the best subset among the evaluated ones is chosen.

Validation Where the selected subset is validated.

4.1.2.1 Major feature selection approaches

Feature selection approaches can be divided into two major categories:

Wrapper methods Wrapping could be carried out using two kinds of algorithms: a search algorithm and a learning algorithm, also called the induction algorithm. The feature selection search algorithm makes use of the learning algorithm, basically a classifier (Ma et al., 2017), to evaluate the selected subset on a held-out dataset. The feature set is updated and the search and evaluation process is carried out again until the selection of

the best feature subset (Kohavi et al., 1997). The resulting learning algorithm could then be used for the target task.

Alternatively, the common technique in CBIR is to use only a search algorithm that maximizes the objective function (Chandrashekar and Sahin, 2014). Instead of using a separate classifier, this function is used to assess the usefulness of the selected subset of features. Wrapper based CBIR frameworks are mainly based on meta heuristic search methods, such as genetic algorithms (Benloucif and Boucheham, 2014), Sequential Forward Selection (SFS) (Belattar et al., 2018), and ACO (Rashno and Sadri, 2017). Another recent and emerging application of wrapping within CBIR are CNNs. These algorithms could be considered as wrappers where the performance is measured in terms of loss (Kabir et al., 2010) or accuracy. This technique will be discussed in Section 4.

Filter methods Compared to wrapper methods, the process of selection and learning is sequential. Moreover, the learning phase is an independent task, making the selection a preprocessing step (Rodriguez-Galiano et al., 2018). Filter methods evaluate a set of features depending on some data characteristics, such as correlation, distance, consistency ... etc (Cherrington et al., 2019). They are then passed to a learning algorithm to perform the desired task.

Examples of Filter based methods in CBIR are Dawood et al. (2019) and Raza et al. (2018), where the correlation between features has been exploited to select the relevant ones.

Compared to wrappers, filter methods are more efficient. The reason is that wrappers evaluate every generated feature subset using a learning algorithm (Tsamardinos and Aliferis, 2003). However, for the same mentioned reason, wrappers often lead to a higher accuracy (Hsu et al., 2011).

4.1.3 Distance Selection (DS)

In fact, the DS paradigm not only refers to the selection of a distance function from a set of distances, it also considers similarity measures, divergences, and quasi-distances (Mosbah and Boucheham, 2017c). All these are referred to as “matching measures” (Mosbah and Boucheham, 2017a).

The choice of the suitable matching measure is a crucial step in CBIR. Recent and early studies demonstrate its impact on the performance of CBIR systems (Mosbah and Boucheham, 2017b) (Pardede et al., 2017) (Vadivel et al., 2003). Based on the NFLT, all matching measures must vary in performance depending on some criterion, including the used image characterization algorithms (Mosbah and Boucheham, 2017b), the database class (Pardede et al., 2017), or even the number of retrieved images (Vadivel et al., 2003). Consequently, it is important to adapt the used matching measure depending on those influencing factors, and here where the distance selection paradigm intervenes.

In this regard, Mosbah and Boucheham (2017a) is one of the few contributions addressing this paradigm by automatically selecting a similarity matching algorithm for each query. To our best knowledge, this paper is the pioneer of the distance selection paradigm within CBIR, where the distance selection using the SFS and relevance feedback algorithm outperformed the best single matching measure in terms of precision and recall.

4.1.4 Relevance feedback technique selection

The aim of the selection of the relevance feedback technique is to select the most adequate technique, depending on an input query (Mosbah and Boucheham, 2015). In this work, the selection was carried out with respect to the number of relevant and irrelevant images returned by each technique, where the selection based technique outperformed the considered relevance feedback techniques.

4.1.5 Threshold selection

In this selection technique, the idea is to consider the threshold as an algorithm parameter and tune it depending on the retrieval task. An interesting relevant work is Mosbah and Boucheham (2016), which selects the most appropriate threshold for the CCV algorithm. The CCV threshold is used to distinguish between coherent and non-coherent regions for color indexing. The selection was based on pseudo-relevance feedback techniques, which means in the runtime, resulting in a dynamic selection.

4.2 Information retrieval tools

We overview two well-known information retrieval tools used in the CBIR area: relevance feedback and query expansion. In this section, we discuss both off them and give some of their applications in CBIR.

4.2.1 Relevance Feedback (RF)

RF is a technique that expands a query in a progressive way based on the user feedback. The user intervention consists in classifying the retrieved documents into relevant or non relevant. This information could then be used to enhance the query and improve the retrieval quality.

The feedback could be carried out using three main approaches (Hong et al., 2000): 1) considering only the relevant samples (Fathian et al., 2018), 2) weighting the relevant samples based on their relevance to the query (de Ves et al., 2016), 3) considering both relevant and irrelevant samples (Tzelepi and Tefas, 2016). One of the most known algorithms from the third category is Rocchio's (Rocchio, 1971) which has been widely used in CBIR (Karamti et al., 2018) (Banerjee et al., 2018) (Xu et al., 2017) (Markonis et al., 2017) (Markonis et al., 2016). In this algorithm, the user is asked to classify the retrieved elements into relevant or non relevant, and based on his feedback, an optimal vector could be generated. The latter is combined with the query vector resulting in a more representative query. The mathematical formulation of the new query vector generation is as follows:

$$Q_1 = \alpha Q_0 + \beta \frac{1}{|R|} \sum_{R_i \in R} R_i - \gamma \frac{1}{|S|} \sum_{S_i \in S} S_i \tag{4.1}$$

Where Q_1 is the new query vector, Q_0 is the initial query vector, , $R = \{R_1, R_2, ..., R_n\}$ is the set of documents selected as relevant, $S = \{S_1, S_2, ..., S_n\}$ is the set of documents selected as non relevant, α, β and γ are the respective weights of the query Q_0, relevant vectors, and non relevant vectors.

Relevance feedback has been applied in CBIR through many techniques, such as Bayesian networks (Glowacka et al., 2016), query point movement (Liu et al., 2008), query expansion (Nguyen et al., 2012), swarm particles (Zhu et al., 2017), and SVMs (Wang et al., 2016). More recently, deep learning (Xu et al., 2017) came into play and drew the path to explore a new technique that has already imposed itself in the CBIR field.

4.2.2 Query Expansion (QE)

This technique consists in merging the query with the selected retrieved elements to make a more powerful query in terms of the corresponding retrieval accuracy.

Unlike the relevance feedback, the QE process is a fully automated process. Moreover, compared to the relevance feedback which biases the query towards the relevant ones, the query expansion aims at bridging the semantic gap between the user expectations and the retrieved elements (Carpineto and Romano, 2012). The bias towards the relevant elements is achieved through combining the query with the selected relevant elements by the user. On the other hand, when expanding a query using the query expansion process, we have no information about the relevance of the elements to be combined with the query.

Carpineto and Romano (2012) explains this for text based retrieval: "the expansion features are related to the full meaning of the query because the extracted terms are those that best characterize the pseudo-relevant documents as a whole, but their association with the query terms is not analyzed explicitly". If we apply this definition on CBIR, the terms refer to the visual features. Based on this, the idea is extract the visual features of the top k retrieved elements, regardless of their relevance to the query.

Despite of these differences, the query expansion technique is strongly related to the relevance feedback as the latter makes use of it to expand the query as in Mohanan and Raju (2017). In addition, the relevance feedback inspired one of the most known Average Query Expansion (AQE) techniques, that is "the pseudo-relevance feedback" (Carpineto and Romano, 2012), which consists of merging the query with the top k retrieved documents assumed relevant (Raman et al., 2010).

This technique is usually referred to as AQE (Chum et al., 2007), and it is the most commonly adopted in CBIR as a postprocessing technique (Gordo et al., 2017) (Xu et al., 2018b) (Pang et al., 2018) (Xu et al., 2018b).

4.3 Deep learning (DL)

In this section, we discuss one of the most recent adaptive techniques, that have been introduced in the CBIR area, that is, deep learning.

Deep learning is a subfield of machine learning that operates on complex data, such as images and voice, for the purpose of extracting relevant patterns from it. In order to achieve this, it performs non linear transformations on input data, using supervised or unsupervised learning methods (Deng et al., 2014). The non linearity is a key aspect for deep learning and is needed when operating on non linear problems. Indeed, it has been shown that it leads to a faster convergence and a better performance (Chollet, 2017).

Deep learning structures are called deep neural networks (DNNs), i.e. neural networks having at least one hidden layer. These algorithms have been successfully applied to many tasks, such as pattern recognition (Ahmed et al., 2015), image classification (Kussul et al., 2017) , sentiment classification (Tang et al., 2014), voice recognition (Buyukyilmaz and Cibikdiken, 2016) and natural language processing (Gardner et al., 2018).

There exists several types of deep neural networks, depending on their architecture, such as Recursive Neural Networks (RNN), Recurrent Neural Networks (RNN), Long short-term memory (LSTM), and Convolutional Neural Networks (CNNs).

4.3.1 Deep learning vs pattern recognition

Deep learning and pattern recognition are two different cognition approaches, sharing the purpose of visual recognition. Lake et al. (2017) considers the learning in pattern recognition as the process of discovering relevant features from the training data, however, the other AI approach is a "model-building" process, where models act just like humans, and are able to think, imagine, understand... etc.

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are an illustrative example of the *model* concept and its ability to perform more operations than pattern recognition, namely imagination. These systems are able to generate new images based on input data, where amazing results have been produced (Mao et al., 2017).

On the other hand, Schalkoff (2007) considers these two techniques as overlapping, since neural networks are one of the pattern recognition techniques, and are considered as "the black box implementation of pattern recognition".

To sum up, these two techniques are strongly related, where the deep neural network aims to recognize hidden patterns in data, resulting in a pattern recognition process, and pattern recognition relies on deep learning as one of the possible techniques to discover data patterns.

4.3.2 Some deep learning based applications

Deep learning has overwhelmed many areas, resulting in a wide range of useful applications. Below, we describe some of them.

Self driving cars These state of the art cars are able to circulate smoothly, detect obstacles and avoid them, and have an accurate understanding of scenes around them (Daily et al., 2017), without any human intervention. As examples, Mercedes-Benz [1] is one of the pioneers in this technology with their luxury autonomous car, Waymo or Google self driving car project [2] aims to facilitate the traffic and save lives, and even Uber [3] is working on integrating a self driving car system into their cars to enhance their service.

Chest X-ray images classification Deep learning has been utilized for disease diagnosis by classifying patients X-ray images, so as to recognize the illness and provide the suitable treatment. A relevant example is COVID-19 diagnosis, where good classification results have been obtained (Hemdan et al., 2020).

Text translation Deep learning is also automating text translation. The well known widely used Google translate service [4] is based on deep neural networks that perform text translation from and to a wide variety of languages.

Video surveillance One of the most interesting applications of video surveillance is anomaly detection in video streams (Xu et al., 2015). A possible scenario for this is to detect the presence of cars on pedestrian streets (Zhou et al., 2019), accidents, roberry, or explosions (Sultani et al., 2018).

Online shopping recommendations An illustrative example of this is Amazon online shopping that recommends products to customers using a deep learning neural network [5], so as to increase its sales.

Fraud detection Due to the increase in the fraud rates, there is a need to strengthen security systems in e-payment transactions. Deep learning has been also useful in this field by detecting fraudulent transactions with a high detection rate (Roy et al., 2018).

[1] `https://mercedes-benz.com/en/innovation/autonomous/research-vehicle-f-015-luxury-in-motion/`
[2] `https://waymo.com/`
[3] `https://uber.com/us/en/atg/technology/`
[4] `https://translate.google.com/`
[5] `https://amazon.science/the-history-of-amazons-recommendation-algorithm`

DeepDream DeepDream was invented to help researchers understand how ANNs encode information (Castelvecchi, 2016). The idea is to feed a neural network an image and a specific pattern, then the NN outputs the image with the input pattern everywhere in the input image (Castelvecchi, 2016). The generated images were fascinating, where sample images could be observed in DeepDream website [6].

Gaming Deep learning has also beat the human brain in many computer games. AlphaGo was a revolution in this field, being "the first computer program to defeat a professional human Go player, the first to defeat a Go world champion, and is arguably the strongest Go player in history", according to its website [7].

It should be highlighted that in our thesis, we have been particularly interested in CNNs, since this architecture is the most used in the CBIR field. It will be discussed in Chapter 6.

4.4 Conclusion

Due to the variety of tasks in terms of the used data, the adaptation could impose itself in CBIR as a tool for improving these systems. Indeed, the adaptation was present through a variety of techniques, where each of them has its pros and cons. In this respect, it should be highlighted that even the adaptive technique could be regarded as an algorithm, where it is up to the system designer to select the more suited to the target task.

[6] `https://deepdreamgenerator.com/`
[7] `https://deepmind.com/research/case-studies/alphago-the-story-so-far`

Part III

Personal Contributions

Chapter 5

Adaptive Content Based Image Retrieval Based on Rice Algorithm Selection Model

In this first contribution, we consider the following decomposition of a CBIR-algorithm: color space, features, similarity matching measure, information retrieval technique, and other potential tuning parameters. Following this, our idea consists in adapting the color space and feature components to the query image, while assigning the other components a predefined value, limiting the degree of freedom to two components per algorithm applied on a specific input image. As such, the resulting portfolio includes a variety of CBIR-algorithms that differ in terms of these two components. This variety enables the usage of the algorithm selection based on the NFLT, with aim to increase the retrieval effectiveness, compared to the single strategy, which is based on the same components for all input images.

5.1 The Algorithm Selection (AS) problem

The AS problem was defined by Rice (1976). We encounter this problem when trying to solve a problem instance having more than one potential solver, which is mainly the case in optimization problems. Based on the NFLT, solvers must vary in performance given a problem instance if their average performances on other problem instances are not equal. In such a case, we take advantage of the performance disparity between algorithms given a specific problem instance by

selecting the best performing algorithm for that instance, which is the concept of the algorithm selection problem. The algorithm selection paradigm has been applied successfully to many tasks such as resource scheduling (Messelis and De Causmaecker, 2014), propositional satisfiability problems (SATZILLA) (Xu et al., 2008), and software model checkers (Tulsian et al., 2014).

5.1.1 Rice basic model

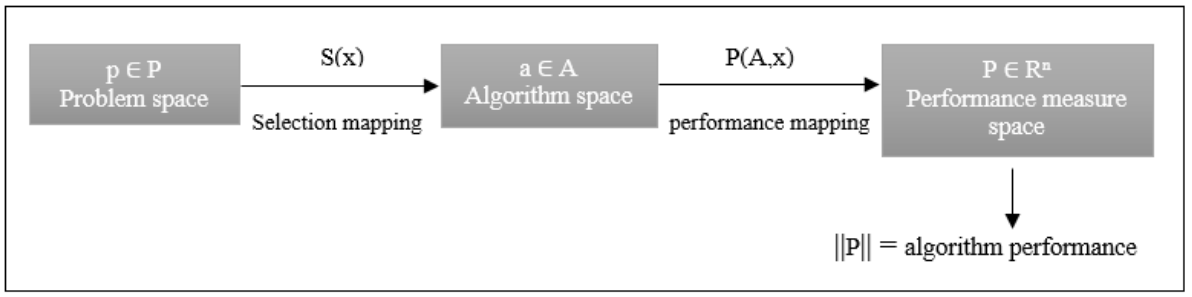


FIGURE 5.1: RICE model for algorithm selection (Rice, 1976).

Figure 5.1 illustrates Rice model for algorithm selection. The model is described as follows:

x Is a problem instance.

a Is a solver algorithm.

p Is the performance of an algorithm a when applied on a problem instance x.

The problem space P Contains the problem instances meant to be solved.

The algorithm space A Contains all the candidate algorithms that will be used to solve the problem space instances.

The performance measure space R^n Contains the performance measure of each algorithm when applied on each problem instance apart.

The selection mapping $S(x)$ Selects an algorithm from the algorithm space to solve a problem instance. Here, it is question of selecting the algorithm that is more likely to perform best on this problem instance.

The performance mapping $P(A, x)$ Maps each algorithm applied to a specific problem instance to its performance measure. The performance measure depends then on both the algorithm and the problem instance.

To summarize the figure above, the algorithm selection process is as follows: First, each algorithm is mapped to one of the problem instances. After that, another mapping between the problem instance solver and its performance is carried out in order to measure how good is that algorithm for that problem instance. Finally, for each problem instance, the algorithm with the best performance measure is selected.

Generally, we try to select the best performing algorithm for each problem instance. This type of selection is called "Best selection" and is one of the six main types of selection defined by Rice (1976):

Best selection As previously mentioned, in this type of selection, the best performing algorithm a is selected for each problem $x \in P$.

Best selection for a subclass of problems As indicates its name, the selected algorithm a is the best performing on all members of a subclass of problems $P_0 \in P$.

Best selection from a subclass of mappings In this case only a subclass from the algorithm space $A_0 \in A$ is involved in the selection mapping. The idea is to select the subclass of algorithms A_0 which minimizes the performance degradation for all problem instances P.

Best selection of mappings and problems This type of selection is a combination of the two previous selection types: First, a subclass of algorithms $A_0 \in A$ is selected, then an algorithm a is selected from this subset to solve a subset of problems $P_0 \in P$.

5.1.2 Rice extended model

This model considers an additional important step in the process of selection, which is "feature extraction". The features are extracted from the problem space, and depending on those features, the algorithm will then be selected. Smith-Miles (2009) defines this model as follows:

"For a given problem instance $x \in P$, with features $f(x) \in F$, find the selection mapping $S(f(x))$ into algorithm space A, such that the selected algorithm $a \in A$ maximizes the performance mapping $y(a(x)) \in Y$".

According to Rice, the main novelty of this model is that the selection mapping depends on the features instead of problem instances. Thus, these features must be representative for

each problem instance so that the selection is adequate. If the extracted features are not representative, no selection strategy would be efficient (Loreggia et al., 2016). Generally, the feature extraction process is not done automatically, but requires a domain expert knowledge (Xu et al., 2008).

Figure 5.2 illustrates Rice extended model taking into account the feature extraction process:

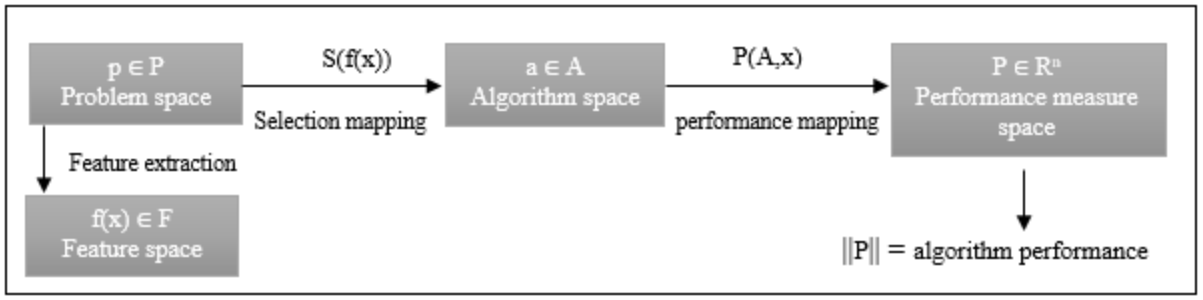


FIGURE 5.2: RICE extended model for algorithm selection (Rice, 1976).

Compared to Rice basic model, the two extensions are:

The feature space F Contains problem instances features $f(x)$.

The mapping from P **to** F Maps each problem x to its features $f(x)$.

The rest of the model remains unchanged.

5.1.3 Contemporary model

Figure 5.3 illustrates the contemporary model for algorithm selection as defined by Kotthoff (2016):

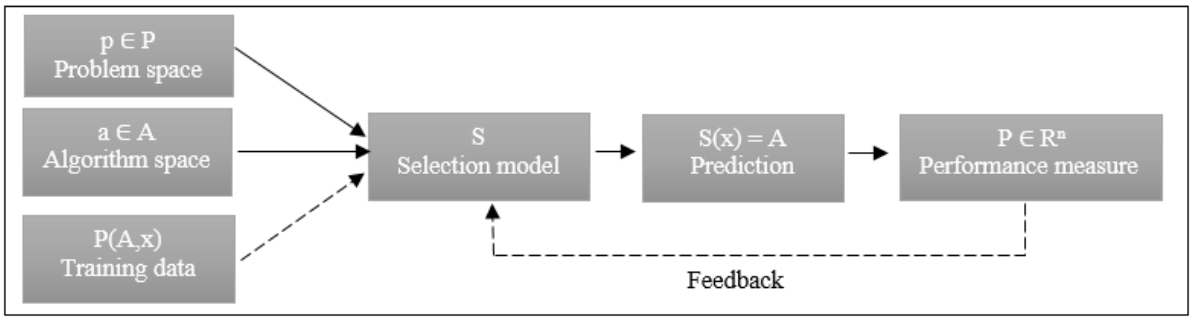


FIGURE 5.3: Contemporary model for algorithm selection (Kotthoff, 2016).

In the figure above, the training data is the result of a training step. In this step each algorithm a is applied to each problem instance x to measure its performance p. This information is saved to be used later by the algorithm selector. The algorithm selector is a *special* algorithm that is used to predict the best performing algorithm on each problem instance. In order to achieve this, two options are available:

Training data : The algorithm selector uses the training data to select an algorithm a for a new problem instance x.

Feedback : The algorithm is applied to a problem instance x in order to measure its performance p.

5.1.4 Algorithm Portfolio (AP)

The AP concept was defined by Huberman et al. (1997). A portfolio is composed by candidate algorithms to solve a specific problem instance. Consequently, it is strongly related to the algorithm selection problem. The reason behind using an algorithm portfolio is increasing the overall performance, i.e. according to Xu et al. (2008), the candidate algorithms "exploit the lack of correlation in the best-case performance of several algorithms in order to obtain improved performance in the average case", which is also the aim of the algorithm selection.

Algorithm portfolios can be classified depending on two criterion: the way of running algorithms, and how the running time is shared between them (Calderín et al., 2015). The first criterion divides Portfolios into two categories: sequential and parallel. The sequential portfolio is based on the algorithm selection technique, ie. it uses a training phase to sequentially run solvers on training instances in order to select an algorithm for each new problem instance (Lindauer et al., 2015). In the parallel portfolio, for each problem instance, algorithms are run concurrently on different processing units (Calderín et al., 2015). Herein, the running time is the algorithm performance measure. Thus, Parallel execution allows solving an instance in the shortest possible delay.

Two well known applications of sequential and parallel portfolios are respectively: SATzilla (Xu et al., 2008) and ppfolio (Roussel, 2012), which have been proposed to solve SAT problems. Satzilla is based on the construction of a portfolio that contains uncorrelated algorithms in terms of their running times. The latter is then measured on a validation set and stored to be employed in the online phase where the algorithm that is likely to perform best on a given

problem instance is selected. On the other hand, the concept of ppfolio is as follows: The best potential solvers are chosen from the last SAT challenge, solvers are then selected to take part of the portfolio based on their ability to solve several instances within a time interval. In both works, the fastest solver for each SAT instance is selected.

The second criterion, which depends on the running time sharing, classifies portfolios into static portfolios and dynamic portfolios. In the static portfolio, the running time is shared between algorithms offline. Moreover, the algorithms are built offline and the portfolio remains unchanged. In the dynamic portfolio, the running time is allocated during solving (Calderín et al., 2015), and the algorithm portfolio may change by adding new algorithms or by changing their settings (Kotthoff, 2016). Moreover, dynamic portfolios use pre and backup solvers. While pre-solvers are used to solve easy instances that do not require a feature extraction process in order to accelerate the running time, backup solvers are used in case of time out while solving a given instance (Rizzini et al., 2017). As an example, Satzilla prepares pre and backup solvers in the portfolio construction phase. Thus, it falls in this category.

5.1.5 Building the empirical hardness model and making use of it

The empirical hardness model is the resulting data from training an algorithm selection tool. This data is used to select the best algorithm for each problem instance. It consists of triplets of data having the following form: (problem instance features, algorithm, performance measure). Our methodology for the construction and the deployment of this data is based on the method presented in Xu et al. (2008), and it is as follows:

After deciding the algorithm space components and the training instances, as well as the criterion to select the algorithm (the performance measure), the learning phase follows these steps:

1. Extract representative features $f(x)$ from each problem instance x.
2. Apply each algorithm a from the algorithm space A on each problem instance x represented by its features $f(x)$.
3. Calculate the performance measure of algorithms R^n: Give the performance p of each algorithm a for each problem instance x.
4. Organize the obtained information to make triplets of data: (problem instance features, algorithm, performance measure).

In the deployment phase, When we desire to select an algorithm for a new problem instance, we make use of the built empirical hardness model. After extracting features from that instance, a comparison is established between the new problem instance features and database instances features. Our assumption is that problem instances with similar features are more likely to have the same solver algorithm. Thus, the algorithm selector compares the new problem instance features to the stored problem instances features, and based on the similarity between features, a solver is selected.

5.1.6 When to opt for an algorithm selection tool?

After all, it is necessary to get back to the starting point: There is No Free Lunch. As any other algorithm, an algorithm selection tool might perform well in some cases, and be useless in others. Consequently, it is necessary to limit the scope of its applicability by underlying the required conditions of its usefulness. According to Messelis and De Causmaecker (2014), there are two conditions that need to be satisfied in an algorithm selection tool: Competitiveness, and potential impact. Competitiveness means that each algorithm performs better than the rest of the portfolio on at least one problem instance. In addition, the potential impact is the degree of competitiveness, in other words, it is the difference in the performance between algorithms on a specific problem instance where higher is better. Thus, a useful algorithm selection tool must be based on a highly competitive portfolio.

5.2 Our proposed approach

In the proposed approach, The training phase is based on RICE extended model (see Figure 5.2), and the deployment of the selection tool stands on the contemporary model (see Figure 5.3).

5.2.1 The algorithm selection training phase

The application of RICE extended model on the training phase is as follows:

The problem space P : Contains the training images from the used dataset. Data splitting configuration is as follows: 60% of images are used for training, while the remaining 40% are used for testing.

The feature space F : Contains the features extracted from the training images. As already mentioned, it is necessary to choose representative features . After experimenting with several features, we ended up choosing color moments since it was the most representative feature among the experimented ones. Herein, color moments is regarded as a feature to represent the problem space. However, in the algorithm space it is regarded as an algorithm that takes part of the portfolio. Exceptionally, in our case features and algorithms have the same goal: problem space characterization, which is the reason behind the overlap between the algorithm space and the feature space.

The algorithm space A : The algorithms are the features to be used for images characterization. Our portfolio is composed of six algorithms, which are couples of the used color space and the feature itself: (RGB, color moments), (HSV, color moments), (RGB, color histogram), (HSV, color histogram), (gray level, LBP), and (gray level, GLCM). The portfolio has been built carefully: After experimenting with several algorithms, we selected only the above mentioned algorithms that were competitive enough to take part of the portfolio.

The performance measure space R^n : The performance measure in this case is the precision of each algorithm when it is applied on each training image.

After calculating the performance measure, the empirical hardness model could be acquired by mapping each training image to its best algorithm (the algorithm having the highest precision on that image), and mapping this algorithm to its precision. After that, We can obtain data having the following form: (image color moments, best algorithm, precision). The training data in this case is used in the query algorithm selection process.

5.2.2 CBIR training phase

In this phase, all algorithms are used to characterize training images resulting in six training subsets. Each subset contains the training images represented by one of the candidate algorithms. The reason behind this is that only images represented by the same features are comparable: Once an algorithm a is selected for the query, the comparison in the retrieval phase must be done between the query and the training images, both represented by the same algorithm a. The resulting training data in this case is used in the CBIR retrieval phase.

5.2.3 The algorithm selection testing phase

The algorithm selector is the main actor in this phase. Its role consists in choosing the algorithm that is more likely to have the highest precision on a given query image (the new problem instance). The flowchart below illustrates the selection process as well as the retrieval phase:

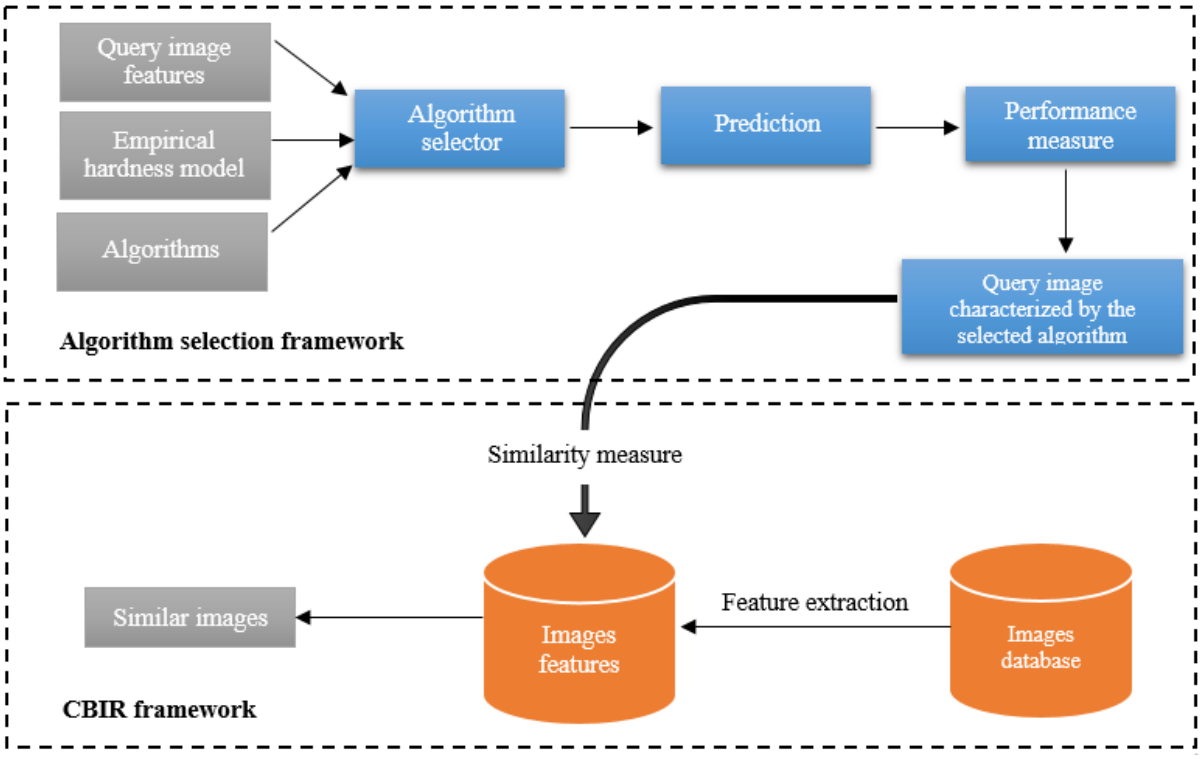


FIGURE 5.4: Our proposed approach including the algorithm selection and CBIR frameworks.

The necessary inputs for the selection process are:

Query image features $f(x)$: The concept of the algorithm selector is based on the comparison between the query features and the empirical hardness model features (RGB color moments). Thus, the query must also be represented by RGB color moments.

Training data (F, A, R^n) : Consists in the already built empirical hardness model.

Algorithms A : The same algorithm space used in the training phase, that is: (RGB, color moments), (HSV, color moments), (RGB, color histogram), (HSV, color histogram), (gray level, LBP), and (gray level, GLCM).

Using these inputs, the algorithm selector predicts the best algorithm for a query image by extracting the k nearest neighbors to that image, as well as their respective best algorithms

FIGURE 5.5: Some samples representing the 10 semantic classes of Wang database.

from the empirical hardness model. After that, a majority voting is applied to select the most adequate algorithm for that query [1].

5.3 Experiments and results

The experimented dataset is Corel-1000 (see Chapter 2). Based on our experiments, k - which represents the number of nearest neighbours - is an important parameter for the selection tool. We have considered several k values and checked the evolution of the retrieval performance. Results have shown that the highest precision was reached with $k = 50$ as shown in 5.6.

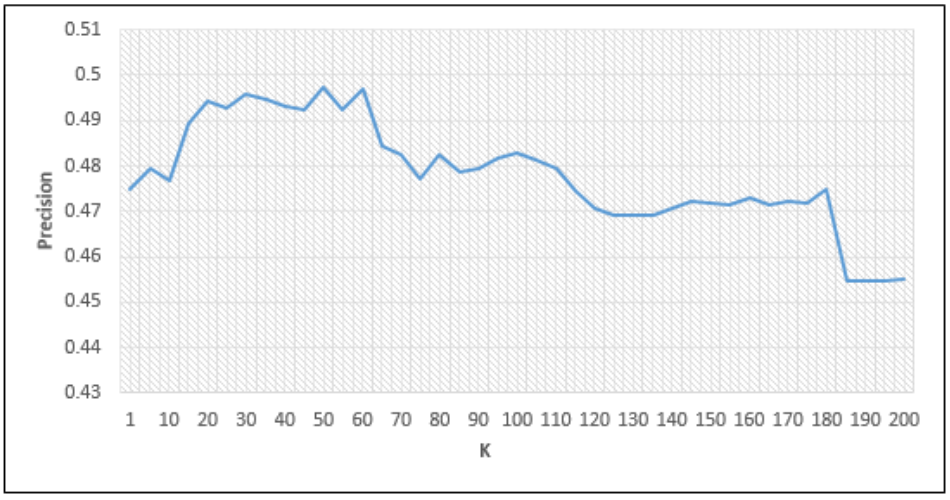


FIGURE 5.6: Precision of the algorithm selection tool in terms of the parameter k.

[1] We found later that a similar KNN approach for algorithm selection was also proposed in Lindauer et al. (2015). The approach is labeled "Distance-based Nearest Neighbor (DNN)"

Finally, taking into account the selected algorithm a, the similarity between the query and the training images, both characterized by a, is measured using the Euclidean distance. The top similar images to the query are then retrieved.

The algorithm selection tool is judged useful if it outperforms the best single strategy. In our case, the latter is (RGB, color moments). Table 5.1 shows the performance of each algorithm apart as well as the performance of the algorithm selection tool. We can see that our tool outperformed the best single algorithm by 4% which is enough to justify the usefulness of the built approach in CBIR.

	Africa	**Beach**	**Bus**	**Dinosaur**	**Elephant**	**Flower**	**Food**	**Horse**	**Monument**	**Mountain**	**All**
(RGB, color moments)	0.3675	**0.3375**	0.4025	**0.9925**	0,3725	**0.4675**	0.3825	0.6975	0.23	0.2975	0.45475
(HSV, color moments)	0.3025	0.2875	0.53	**0.9925**	0,3025	0.2575	0.4	0.69	**0.2875**	**0.36**	0.4465
(RGB, color histogram	0.4575	0.195	0.4325	0.7575	**0,4575**	0.335	0.365	0.575	0.1675	0.2725	0.4115
(HSV, color histogram)	**0.5825**	0.1775	0.6075	0.75	0.2375	0.3225	0.4	0.87	0.155	0.3175	0.442
(gray level, LBP)	0.1925	0.2475	**0.74**	0.945	0,1925	0.435	0.1875	0.4	0.2725	0.195	0.393
(gray level, co-occurence matrice)	0.335	0.1375	0.23	0.9725	0,335	0.4325	0.2875	0.3575	0.0925	0.14	0.3325
Algorithm selection tool	0.5225	0.2975	0.5275	**0.9925**	0.3275	0.46	**0.42**	**0.86**	0.2575	0.3075	**0.4972**

TABLE 5.1: A comparison between the performance of candidate algorithms against the algorithm selection tool in terms of precision on Corel 1k dataset, best results are highlighted in **bold**.

TABLE 5.2: Competitiveness between candidate features expressed in terms of the number of images having each algorithm as the best performing algorithm.

Algorithm	Degree of competitiveness
(RGB, color moments)	68
(HSV, color moments)	52
(RGB, color histogram	25
(gray level, LBP)	04
(HSV, color histogram)	37
(gray level, co-occurence matrice)	14

5.4 Conclusion

We could demonstrate that the algorithm selection technique is useful for CBIR. Despite this, we should admit that the performance of our system was still low, compared to the state of the art. The reason is that the employed algorithms are very basic and the performance of the developed tool strongly depends on the candidate algorithms. A possible improvement could be strengthening the portfolio by including robust algorithms for image characterization, which will consequently enhance the performance of the selection tool. It would be also interesting to increase the degree of freedom to enlarge the portfolio and raise the diversity degree, by varying more components in the CBIR-algorithm. This results in a more flexible CBIR system, since more components are adapted for input images.

(a) Query

(b) Relevant images retrieved using the best single strategy approach

(c) Relevant images retrieved using the algorithm selection framework

FIGURE 5.7: An example of a query image from Horse class and retrieved images using both the AS and the best single strategy approaches. The image from the first row represents the query and the remaining images represent the retrieved images.

Chapter 6

Content Based Image Retrieval by Convolutional Neural Networks

CNNs are one of the most omnipresent manifestations of machine learning in CBIR. These models intervene in the indexation phase of the CBIR framework aiming at the extraction of discriminative features. Our contribution consists in the employment of a pretrained CNN to extract useful features from the experimented dataset. Moreover, we expand the experiments to the evaluation of the two already presented location dependent query types to investigate their impact on the CBIR performance.

6.1 Convolutional neural networks

CNNs are a family of deep learning neural networks, that have been successfully applied to many computer vision tasks (Li et al., 2015), (Li et al., 2014) (Molina-Cabello et al., 2018) (Fu et al., 2016a) (Long et al., 2017), including image retrieval by content (Wan et al., 2014a) (Babenko et al., 2014) (Noh et al., 2017) (Gordo et al., 2017) (Wei et al., 2017). Figure 6.1 illustrates some examples of computer vision tasks performed by a CNN. Two reasons are behind the recent widespread of this type of DNNs: the computational power embodied in Graphical Processing Units, and the availability of data, i.e. large scale image datasets that are needed to train these architectures.

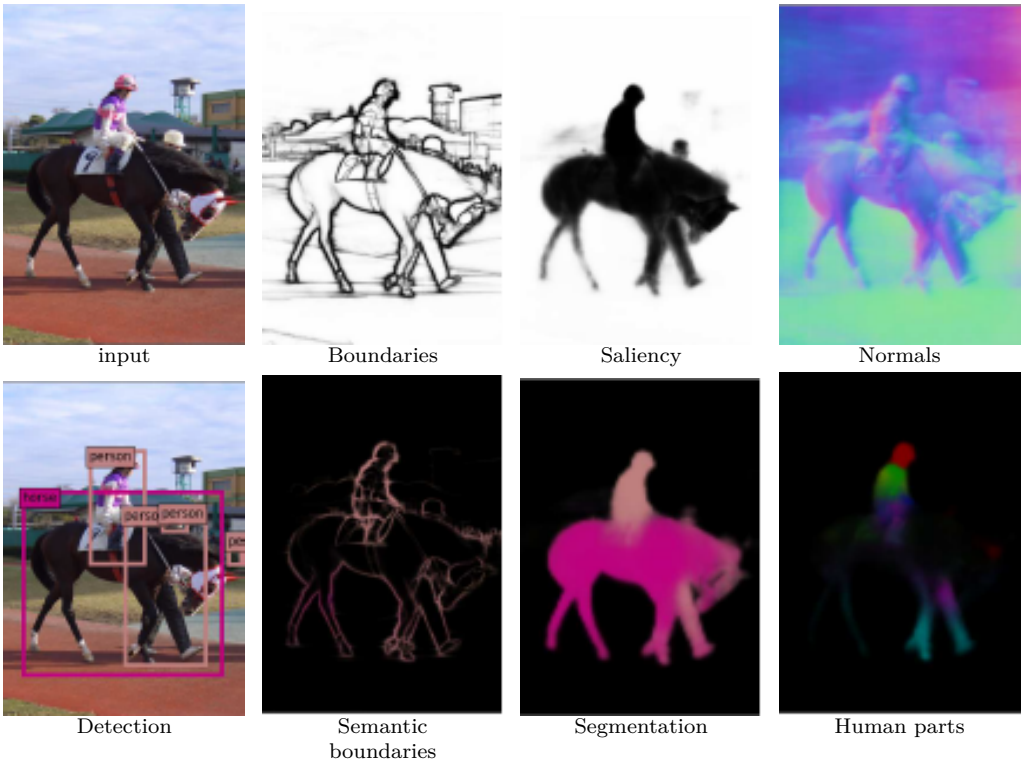


FIGURE 6.1: An input image and the result of applying several computer vision operations on it using a CNN (Kokkinos, 2017).

6.1.1 Architecture and functioning

Compared to other deep learning architectures, CNNs are characterized by special layers, called "Convolutional layers", the role of which is to extract patterns from images. Each convolutional layer extracts these patterns in a form of feature maps, where each feature is localized in the image. Convolutional layers may be followed by "Pooling layers" to reduce the dimension of feature maps. After that, these feature maps are passed to the subsequent layer, until the image is input to the last layer, i.e. the fully connected layers that classify this images into one of the possible categories.

The extraction of feature maps is performed using a filter of a specific size. This filter is convolved throughout the input image until all the image is processed. The name of this operation is the convolution, hence the name of these layers. This operation is depicted in Figure 6.2, where a filter of size 3*3 is convolving throughout the image.

As previously mentioned, a convolution operation may be followed by a pooling operation, whether max-pooling or average-pooling, as shown in Figure 6.3. This operation first splits the input into regions, where a common region size is 2*2 (Albawi et al., 2017). The maximum

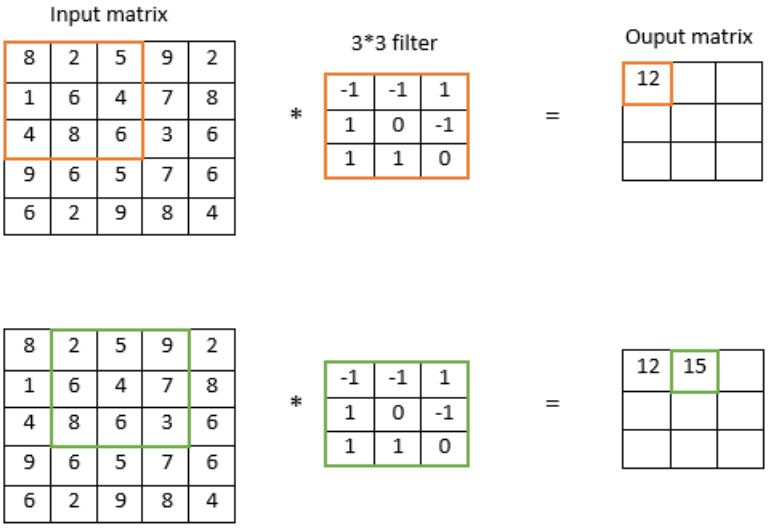


FIGURE 6.2: A convolution operation where the filter size is 3*3. The result in the output matrix is obtained by summing up the lines products of the 3*3 slice of the input matrix and the convolving filter.

value is chosen from each region and stored in a feature vector, which is then passed to the next layer in the CNN.

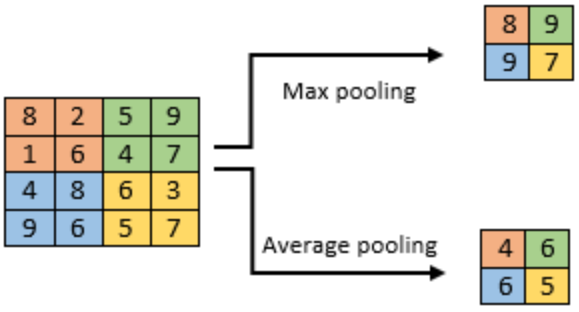


FIGURE 6.3: Average and Max pooling operations.

6.1.2 Training a convolutional neural network: things to consider

There are several considerations to take into account when training a CNN, whether the model is fully trained or finetuned. The first thing to consider is to feed the CNN enough and diverse data, so it is able to recognize different patterns. One of the common used techniques to achieve this is called "data augmentation". This technique generates new samples based on the training dataset samples, by applying several transformations on images, such as rotation, shifting, and zooming. In Perez and Wang (2017), the authors conduct experiments to check the effect of

data augmentation on training a neural network. The experiments showed that generating augmented images to feed the network enhanced the accuracy significantly.

Training a neural network is an iterative process. The same training data is passed through the CNN several times, until it is properly trained. An iteration of the full training data in a neural network is called "epoch". In other words, in a single epoch, all the training data have passed once through the CNN, thus, each sample has collaborated in updating the CNN weights.

The CNN is trained in manner that a defined number of data samples, called "batch size" is passed through the CNN, until all the training set is passed (a full epoch). Thus, The required number of batches to pass all the data through the CNN at least once, depends on both the training data size, and the batch size. The number of epochs is another parameter that has an important effect on the CNN accuracy, where more epochs than required may overfit the model, and less epochs than required will underfit the model. Overfitting is a phenomena that occurs when the model performs well on the training set, but badly on new unseen data. We say that the model depends on the training set, and is not able to generalize, i.e. it has a high variance. On the other hand, underfitting occurs when the model performs badly on both training and unseen samples, i.e. the model has a high bias.

A good model has low bias and variance. There are several techniques to achieve this. The most common one is the use of a validation set while training the model to check its accuracy on the new unseen data in the real time training. If both training and validation accuracies are low, the model is underfitting. However, if over epochs, the training accuracy is improved, and the validation accuracy is worsen, the model is overfitting.

Another technique consists in the use of "dropout layers". These layers have a number of deactivated neurons (their corresponding weights are set to 0), which is also called the dropout ratio. This technique helps to prevent overfitting by decreasing the model's complexity. Last but not least, data augmentation is also beneficial in this regard, by providing more samples to the CNN, which decreases its dependency to the original training data.

To sum up, the training phase in a CNN is of a huge importance, and creating a powerful CNN is back to training it properly.

6.1.3 Pretrained convolutional neural networks

Creating a state of the art CNN architecture is a highly complex task and requires a strong knowledge in deep learning. Moreover, depending on the training set size and the computational power, training a whole CNN may be expensive in terms of computational cost. That's why, pretrained CNNs are usually used in deep learning applications (Olmos et al., 2018) (Gruber et al., 2017) (Chen et al., 2018) (Habibzadeh et al., 2018) (Sugianto et al., 2018). A pretrained CNN is a CNN model that has been already trained on a large scale image dataset, such as ImageNet. As already mentioned, this dataset includes 1.2 million training images, grouped into 1000 categories, allowing the CNN to recognize a large bunch of patterns for the purpose of reusability. When given a new image dataset, pretrained neural networks may be applied in two different ways, whether by extracting the features directly from the pretrained CNN (Hoang et al., 2017), or by training only a few number of the last layers on the given dataset (Pang et al., 2018), or a dataset from the same domain (Noh et al., 2017), in order to adapt the CNN parameters to it. After that, images features could be extracted.

There exists several architectures of pretrained neural networks, the most commonly used in CBIR are VGG (Simonyan and Zisserman, 2014) and ResNet (He et al., 2016a).

6.1.4 Are black boxes worth trust?

There has been a Controversy over trusting AI models including CNNs in recent years. According to Lipton (2018), if the trust is the guarantee that a model will perform well, then the good functioning involves trust. Trusting a model is back to its interpretability. In fact, there is no clear definition of the interpretability in the literature, but several definitions have been made to better clarify this concept:

Definition 1 "The degree to which a human observer can intrinsically understand the cause of a decision" (Feldman et al., 2019)

Definition 2 "Interpretability refers to the extent of human's ability to understand and reason the model" (Fan et al., 2020)

Definition 3 "An interpretation is the mapping of an abstract concept (e.g. a predicted class) into a domain that the human can make sense of" (Montavon et al., 2018)

In addition, interpretability is related to two different concepts (Silva et al., 2019) (Lipton, 2018): "transparency, asking how the model works", which is related to the model's output, and "what else the model can tell", which is related to the system's architecture.

There exists two types of interpretability (Du et al., 2019):

Intrinsic interpretability Models belonging to this category have transparent architectures, making them explainable to humains. However, this transparency is offered on the expense of their accuracy, models example are decision trees.

Post-hoc interpretability The models requiring this type of interpretability have complex architecture, consequently, they need another model to explain their behaviour. DNNs including CNNs belong to this category. Unlike the first category, these models are more accurate due to their complexity, which means that the accuracy is offered on the expense of interpretability.

There has been a large interest in CNNs understanding in recent years (Shang et al., 2016) (Yu et al., 2020) (Jacovi et al., 2018) (Lin and Maji, 2016) (Zeiler and Fergus, 2014). However, there is still no complete understanding of CNNs functioning. One of the main reasons of this is the complex architecture of these models, particularly, the non linear operations with millions of parameters are the source of the algorithm complexity (Fan et al., 2020). Moreover, CNNs are characterized by their depth which reinforces their accuracy. With all those parameters and layers, it is highly difficult to tract the operations performed on the whole image, and how to correlate the final output to an input that has already been subject to a lot of non linear transformations, highly parameterized.

Silva et al. (2019) beautifully links the interpretability with Google photos application that classified black people as Gorillas. No offense was meant, however, due to the inability to explain why the system was behaving in a "racist" way, no solution could be provided to solve the issue. Alternatively, the class Gorillas was removed from the recognition affecting its completion [1]. The link made to the interpretability is that this story clearly reflects the inability to understand black boxes, and also emphasises the interpratibility as an important feature to guarantee the good functioning of systems.

The interpretability is mainly required in decision making systems (Silva et al., 2019), especially when it is a matter of survival or mortality. A relevant example is medical diagnosis: If the

[1] https://www.theverge.com/2018/1/12/16882408/google-racist-gorillas-photo-recognition-algorithm-ai/

black box predicts a patient as safe while he suffers from a critical disease that needs an urgent intervention, then we need to understand how the model is miskating this critical decision in order to correct its behaviour. Another relevant example given by Feldman et al. (2019) are self driving cars that may lead to death if a wrong decision was made. All these examples are in favor of the need of interpretable models, rather than just good models.

As humans, we are naturally comfortable with explainable outcomes. Hopefully, in the coming years we may solve the mystery buried in these black boxes. Until then, the good outcome of these models is our only source of trust. To conclude with a beautiful quote: "You use your brain all the time; you trust your brain all the time; and you have no idea how your brain works"- Pierre Baldi.

6.1.5 On the computational power required by CNNs

CNNs are known to be too demanding in terms of computational cost compared to other visual feature extraction algorithms. The evolution of material depicted in graphical processing units (GPUs) was behind their success in recent years. Indeed, thanks to GPUs we were finally able to transformn CNNs from theoretical models to production models.

The time required by a CNN to be properly trained depends on two main reasons: the image dataset size, where the larger the dataset, the more iterations are required, and the CNN depth, where deeper CNNs are more demanding, since more operations are performed. In fact, some pretrained architectures, namely RESNET (He et al., 2016a) and DensNet (Huang et al., 2017) have been designed to reduce their depth using shortcut layers which decreases their complexity and hence, their training time.

In addition to GPUs, Google has recently unveiled a powerful architecture, that is Tensor Processing Units (TPUs), which are the new competititor of GPUs. The power of these architectures, i.e. both GPUs and TPUs, is mainly exploited in the training phase, where complex operations are performed on images. In this respect, it should be highlighted how the available computational power is increasing over time, as shown in Figure 6.4.

[2] `https://openai.com/blog/ai-and-compute/`

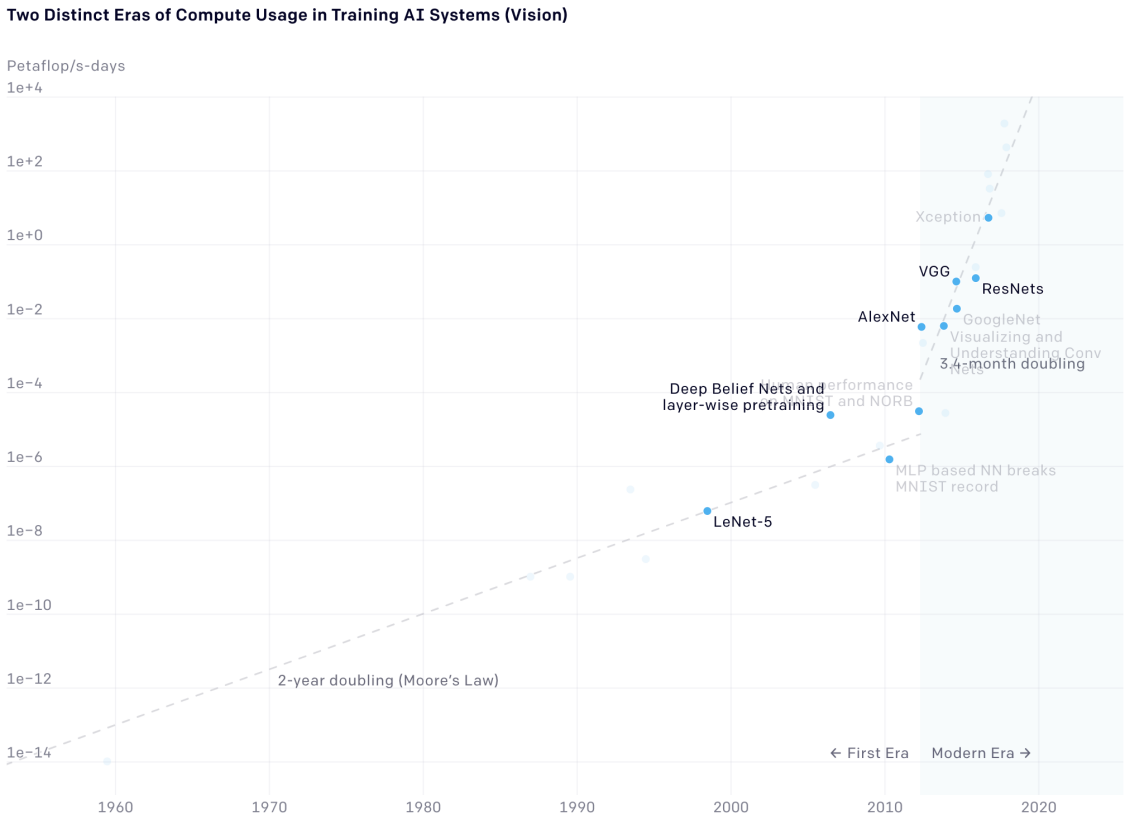


FIGURE 6.4: The available computational power expressed in petaflop/s-days (10^{15} neural net operations per second for one day), used to train some of the state of the art CNN architectures. In the first era, we clearly see that the computational power is increased with respect to Moore's law. In the modern era, the increase is more important and follows an exponential growth [2].

There has been also put an effort to accelerate the inference phase by Intel Corporation through their optimization library OpenVino [3], which was designed to optimize CNNs to run them faster on Intel architectures.

In addition to this, many AI platforms have been released, making robust GPUs more available for researchers and engineers. Examples of these platforms are Google Colab [4], Intel devCloud [5], Google AI platform [6], Amazon AI platform [7], and Microsoft Azure AI platform [8].

[3] https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit.html/
[4] https://colab.research.google.com/
[5] https://software.intel.com/content/www/us/en/develop/tools/devcloud.html/
[6] https://cloud.google.com/ai-platform/?hl=fr/
[7] https://aws.amazon.com/fr/machine-learning/
[8] https://azure.microsoft.com/en-us/overview/ai-platform/

6.2 Convolutional neural networks in the CBIR literature: an overview

One of the pioneers of deep learning in CBIR is Babenko et al. (2014). This paper investigates the effectiveness of CNNs in the area of CBIR, where the conducted experiments revealed that adapting a CNN to the target task by finetuning it on a similar dataset enhances the CBIR performance. Moreover, reducing the vector size using PCA algorithm led to an increase in the discriminative power of extracted features. Indeed, many of the subsequent contributions follow the same process by extracting features from a finetuned CNN and reducing them with PCA reduction algorithm (Radenović et al., 2016) (Wei et al., 2017) (Xu et al., 2018b) (Radenović et al., 2018) (Pang et al., 2018).

More recent contributions follow a common direction, which is the enhancement of extracted deep convolutional features. One of the common techniques is the selection of the most representative regions in the image for the purpose of extracting useful features from them (Xu et al., 2018b) (Xu et al., 2018b) (Wei et al., 2017).

Speaking of reducing the computational cost, a relevant contribution is Mohamed et al. (2017). This paper proposes a CNN SVM based architecture for image retrieval, that could be trained on a CPU. The idea was to perform the feature extraction process once per an image using a CNN, the extracted features are then passed to a SVM to learn them. Moving the training phase from the CNN to the SVM allows to speed it up, since SVMs are much less demanding in terms of computational cost. Moreover, this hybrid architecture led to significant retrieval results.

6.3 Our proposed approach

In order to design a Content Based Image Retrieval (CBIR) system based on a Convolutional Neural Network (CNN), we propose to employ a neural architecture which has an output layer with one output neuron for each object class in the training image database. Let D be the number of object classes. The CNN has to be trained on the images of the training database, where the desired output for each image is a unit vector of size C with a one at the component associated to the class of the object which is depicted in the training image, and zeros at the

rest of the components. Given this configuration, the CNN learns to estimate the probabilities that an image represents an object class:

$$f(\mathbf{X}) = (P(C_1|\mathbf{X}), ..., P(C_D|\mathbf{X})) \in [0,1]^D \tag{6.1}$$

where $\mathbf{X}$ is an input image, and C_i is the i-th object class, with $i \in \{1, ..., D\}$.

After the CNN is trained in this way, the system is ready to accept user queries. Let us note $\mathbf{X}_{Query}$ the query image, and $\mathbf{X}_j$ the training database images, where $j \in \{1, ..., N\}$ and N is the number of images in the database. Then the database images are ranked according to the Euclidean distances between the probability vector $f(\mathbf{X}_{Query})$ associated to the query image, and the probability vectors $f(\mathbf{X}_j)$ associated to the database images. For example, the most similar image in the database can be obtained as follows:

$$s = \arg \min_{j \in \{1,...,N\}} \|f(\mathbf{X}_{Query}) - f(\mathbf{X}_j)\| \tag{6.2}$$

It is then possible to obtain the k most similar images in the database, as ranked by $\|f(\mathbf{X}_{Query}) - f(\mathbf{X}_j)\|$. Alternatively, a similarity threshold τ can be defined so that all images in the database with $\|f(\mathbf{X}_{Query}) - f(\mathbf{X}_j)\| < \tau$ are declared to be similar to the query image.

6.4 Experiments and results

In this section, we report the obtained results by our approach, which uses Convolutional Neural Networks, and compare them with other state of the art traditional approaches.

6.4.1 Methods

In this work, the transfer learning technique using Alexnet pretrained network (Krizhevsky et al., 2012) has been applied. Our work consists of fine-tuning this network to adapt it to our used database by replacing the final layers, we can then train the network so that it learns feature extraction for our new task. As a result, the CNN will be able to extract class probability vectors from database images. This information can be deployed later in the image retrieval task, where similarity between images is calculated based on the distance between class membership probabilities of the query and database images.

The selected methods to compare the results of our proposal are *Walia et al.* (noted as Walia) Ekta Walia, Aman Pal (2014), *Elalami* M.E. ElAlami (2014), *ODBTC* Jing-Ming Guo, Heri Prasetyo (2015), *DDBTC* Jing-Ming Guo, Heri Prasetyo, Nai-Jian Wang (2015) and *LMCTP* Yesubai Rubavathi Charles, Ravi Ramraj (2016). Walia method is based on a combination of color, texture, and shape features using different fusion techniques. The Elalami method introduces an effective matching strategy in order to measure similarity between images. Both of ODBTC and DDBTC consist in generating a feature vector derived respectively from low complexity-Dither Block Truncation and Dot-Diffused Block Truncation Coding. Finally, a merged color space is created in LMCTP, from which local mesh color features are extracted.

6.4.2 Dataset

In this paper, we use Corel 1k dataset (see Chapter 2). 70% of images are used to train the neural network, and 30% are left for the test phase.

6.4.3 Hardware and software

The experiments have been established on a machine with Ubuntu 64 bits operating system, 2,9 GiB RAM, Intel Core 2 QUAD CPU with a frequency of 2,40 GHz and NVIDIA GPU. The used programming language is MATLAB R2018a.

6.4.4 Results

we have employed some well known performance evaluation measures. These measures are: Precision at k (P@k), Recall at k (R@k), where k is the number of images to retrieve, Precision given a distance threshold (P_τ), Recall given a distance threshold (R_τ), where τ is the threshold value. They provide a real number between 0 and 1, where higher is better, and they are given in the following equations:

$$P@k, P_\nu = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ retrieved\ images} \tag{6.3}$$

$$R@k, R_\nu = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ relevant\ images} \tag{6.4}$$

In our experiments, We use two location dependent query types for the retrieval phase. The first is KNN search. Table 6.1 shows the performance of the designed CBIR system in terms of P@k and R@k, where k is set to 20. As shown in the table, the precision values are high for all classes. Thus, the overall precision is close to perfection with 0.9573. However, the recall values are low. This low value is not related to the system performance, it is related to the number of the retrieved images k which is considerably lower than the number of available relevant images for each query (70% from 1000 samples is the ratio of the training phase, equally grouped into 10 classes, that is 70 relevant samples for each query). Thus, even for the classes: Bus, Dinosaur, Elephant, Horse, where a perfect precision is achieved (P@20 = 1), the recall is still low. In this case, there are two possible ways to increase the recall value, whether by increasing the parameter k, or by using a threshold value τ for the distance, which allows retrieving more similar images.

The second retrieval type is distance query, where a threshold value τ is used for the distance. The relevant experiment is shown in Table 6.2. It can be observed in the table that both precision and recall are high when measured over all dataset classes. There is a considerable increase for the recall value compared to the previous experiment which is based on KNN search. We refer the reason for this to the retrieval of almost all relevant available images. In the perfect situation, where all relevant images are ranked first, the respective distances between the query image and all relevant images should belong to a range of values, where images having a larger distance than the maximum distance from this range are irrelevant. This way, a distance threshold could be defined by the distance between the irrelevant image that is ranked first and the query image. Thus, all images having a distance less than this threshold are relevant. However, it is difficult to get such a perfect result in CBIR systems. Thus, we aim to select the threshold where most of images having a distance less than this threshold τ are relevant, which means that compared to all possible threshold values, the precision is higher for the selected threshold τ. In order to achieve this, we evaluated 10 possible threshold values ranging from 0.1 to 1, and selected the threshold value that corresponds to the highest precision, where τ equals 0.6 (see Figure 6.5).

It is also worth mentioning, that compared to defining a number of images to retrieve k, using a distance threshold enhanced significantly the recall of the designed system without decreasing its precision, which was slightly enhanced too.

In order to check the precision and recall values in terms of the number of retrieved images k, we varied the parameter k from 10 to 70, which is the number of relevant images for each query.

TABLE 6.1: Precision and recall values of our approach by using a k fixed number of retrieved images, where $k = 20$.

Classes	Africa	Beach	Bus	Dinosaur	Elephant	Flower	Food	Horse	Monument	mountain	Overall
Precision	0.9333	0.9000	1.0000	1.0000	1.0000	0.9667	0.9683	1.0000	0.9667	0.8383	0.9573
Recall	0.2667	0.2571	0.2857	0.2857	0.2857	0.2767	0.2767	0.2857	0.2762	0.2395	0.2735

TABLE 6.2: Precision and recall values using a defined threshold τ for the distance, where $\tau = 0.6$.

Classes	Africa	Beach	Bus	Dinosaur	Elephant	Flower	Food	Horse	Monument	mountain	Overall
Precision	0.9333	0.9000	1.0000	1.0000	1.0000	0.9667	0.9667	1.0000	0.9662	0.8500	0.9583
Recall	0.9010	0.9000	1.0000	1.0000	1.0000	0.9667	0.9667	1.0000	0.9667	0.8338	0.9535

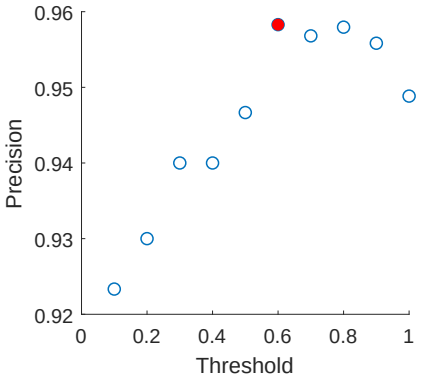


FIGURE 6.5: Precision in terms of some experimented threshold values τ. The red circle indicates the highest obtained precision which equals 0.9583, where the corresponding threshold value τ is 0.6.

Figure 6.3 shows the corresponding experiment. We can observe that the precision is almost fixed for the experimented k values, where even for high k values, the precision is high. However, the recall increases as the parameter k increases, which means that more relevant images are being retrieved.

Generally, precision and recall measures are not correlated, i.e. the increase in the number of images to retrieve enhances the recall, but decreases the precision. The reason is that when more images are retrieved, we have higher chances to retrieve both relevant and irrelevant images. The retrieval of more relevant (true positives) images enhances both precision and recall. However, the retrieval of irrelevant images (false positives) decreases only the precision, without affecting the recall, as explained in Chapter 2. In this respect, the behaviour of the conceived system is different; the recall is improved when more images are being retrieved, while the precision is almost fixed to its high value. The interpretation of this is that most relevant images are ranked higher than the irrelevant ones, which is a strength point of our system.

In addition, Figure 6.7 illustrates the results from a qualitative point of view, where a query

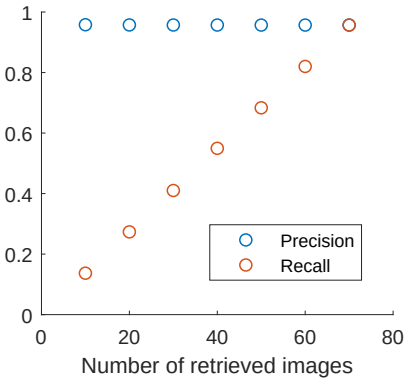


FIGURE 6.6: Precision and recall performance of our proposal depending on the number of retrieved images k.

image from the Flower class, as well as the retrieved similar images are shown. In this example, the parameter k is set to 20, and P@20 equals 1.

FIGURE 6.7: An example of a query image from Flower class and retrieved images using our approach. The image from the first row represents the query and the remaining images represent the retrieved images.

In order to further check the goodness of our proposal, we compare it against the previously mentioned methods. The comparison is shown in Table 6.3. Our method outperforms significantly the compared approaches in terms of P@20.

TABLE 6.3: Comparison between our approach and other relevant approaches in terms of precision. A k fixed number of retrieved images has been used, where $k = 20$. The best result is highlighted in **bold**.

Approach	Precision
Walia Ekta Walia, Aman Pal (2014)	0.783
Elalami M.E. ElAlami (2014)	0.761
ODBTC Jing-Ming Guo, Heri Prasetyo (2015)	0.779
DDBTC Jing-Ming Guo, Heri Prasetyo, Nai-Jian Wang (2015)	0.792
LMCTP Yesubai Rubavathi Charles, Ravi Ramraj (2016)	0.765
Our method	**0.957**

6.5 Conclusion

We could demonstrate that CNN based approaches are very useful in extracting discriminative features from images. Moreover, when evaluated on Corel-1000, the CNN based approach (the implicit feature extraction technique) outperformed some well known approaches, that are based on the explicit feature extraction. Finally, the main advantage of our approach is that even when the parameter k is increased, the precision is stable. In other words, a good ranking could be achieved by the designed CBIR system.

Chapter 7

Content Based Image Retrieval by Ensembles of Deep Learning Object Classifiers

In this section, we extend the previous contribution of this thesis by combining several CNNs that collaborate to extract relevant features from images, that is the ensemble learning paradigm. We extensively evaluate this paradigm within CBIR by conducting experiments on several datasets, and with several ensemble sizes. Moreover, we evaluate two ensemble approaches by using both homogenous and heterogeneous ensembles in terms of the constitute architectures. Similarly to the previous contribution, we also evaluate two location dependent query types: KNN and distance query. Finally, we propose a novel query expansion technique to further enhance the retrieval quality.

7.1 Ensemble learning

Ensemble learning paradigm uses a set of machine learning algorithms to solve a specific problem. These algorithms may be from the same type, i.e. an homogeneous ensemble such as an ensemble of CNNs (Vasan et al., 2020), or from different types, i.e. an heterogeneous ensemble, such an ensemble of CNNs and SVMs (Kang et al., 2017). Diversity is required when building an ensemble, and it can be achieved by using different types of learning algorithms, different architectures of the same learning algorithms, or different training datasets.

The idea behind using an ensemble of algorithms is to construct a set of hypothesis rather than one hypothesis to explain data. These hypothesis are made by the learning algorithms, called "base learners" or "weak learners" (Zhou, 2009). They are then merged in some way to predict the output of a given data sample (Dietterich et al., 2002). Ensemble methods are known to be more accurate than base learners, hence, they have been applied to several computer vision tasks, such as image classification (Rokach et al., 2014), object localization (Markuš et al., 2014), image segmentation (Cyganek, 2012), object detection (Cyganek, 2012), and image reconstruction (Andreyev et al., 2011).

7.1.1 What's behind the power of ensemble methods?

Ensemble learning is related to "The wisdom of the crowd" effect. This means that a group of learners is more intelligent or "wiser" than an individual learner, and is able to make better decisions to solve a specific problem. The diversity of learners in terms of their type, architecture, or training set is the reason of the diversity in their output predictions, which results in the independence of their errors. Thus, an input sample may be wrongly predicted by a base learner, but correctly predicted by another from the same ensemble. Merging the outputs of these two learners, will result in an output which is more biased toward the correct output, compared to the wrong output. Moreover, this output will be improved as the ensemble size grows and the number of base learners that correctly classify this sample grows too.

Technically, the ensemble learning paradigm deals with the bias and variance tradeoff. In machine learning, both bias and variance should be reduced, and ensemble methods are known to reduce bias and/or variance depending on the used ensemble construction method. As an example, the bagging technique (Quinlan et al., 1996) reduces the variance, and is good to apply on models that lack the generalization power, such as neural networks and support vector machines (Zhou, 2009). On the other hand, the boosting technique (Quinlan et al., 1996) reduces both bias and variance and is good to apply on models that learn with difficulty such as decision stumps, i.e. decision trees with one level (Zhou, 2009).

From another point of view, Dietterich et al. (2002) explains why ensemble methods work better than individual learners and refer this to three fundamental reasons. The first one is statistical: The estimation of the best base learner depends on the training data that was fed to the learning models. It is difficult to estimate the best base learner if this data is not large enough. In such a case, the weak learners tend to perform equally. Thus, it is preferable to combine their outputs

to reduce the probability of error. Second, even if the first problem is solved, we still have the computational issue: The learning algorithms are trained in a manner to find the local optima using functions such as Stochastic Gradient Descendant (SGD), and the starting point to search the local optima varies as the learning algorithm varies in terms of type, architecture, or training set. Ensemble methods take advantage of this by combining the outputs coming from learners having different starting points, which may give more accurate predictions. The third and last problem is representational: The problem which is meant to be solved is represented by functions, i.e. the learning algorithms, however, in machine learning applications, a limited number of learning functions is explored, where it is possible that none of them is the right one to apply on the given problem. By combining their outputs, we expand the set of functions to represent the problem, which helps to overcome this issue.

7.1.2 Major ensemble construction methods

There exists several techniques to build an ensemble of machine learning algorithms. The most common ones are bagging, boosting, and stacking. These techniques may use algorithms having the same or different type/architecture. However, they must be trained in a manner to vary the training set with respect to some conditions, as indicated below.

7.1.2.1 Bagging (Bootstrap Aggregating)

This technique consists in grouping the training set in several subsets, called "bags". Each sample in each bag is selected by replacement, i.e. if a sample is selected in one bag, it may be selected again in the same or different bag. Moreover, these bags are of equal size. This way, each bag of data is assigned to a different learner, and the variety is achieved through the variation of the training set for each base learner. In this technique, the classifiers could be trained in parallel.

An example of the application of this technique is Zareapoor et al. (2015), where authors use an ensemble of decision trees to classify credit card transactions into legitimate or fraudulent. The decision trees are initially trained on several bags of data consisting in real life transactions. After that, given an input sample, each decision tree votes for the corresponding class. The final decision is then obtained by weighting the outputs of the component decision trees. The bagging was used to reduce the variance, which led to a high classification accuracy.

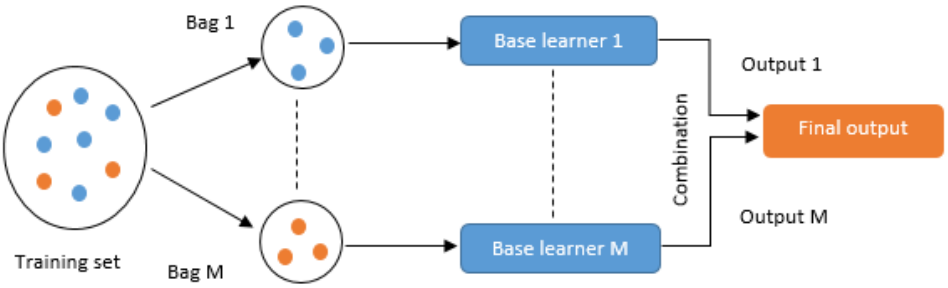


FIGURE 7.1: The bagging technique.

7.1.2.2 Boosting

similarly to the bagging technique, the boosting technique is based on splitting the training set into bags of data to train the base learners. In addition, the boosting technique takes into account the misclassified samples by each weak learner and give them higher impact to train the subsequent weak learner. After a weak learner is trained on a bag of data, the misclassified samples are identified and assigned higher weights to create the following bag to train the next learner, so that these samples have higher impact in training the following classifier (Quinlan et al., 1996).

An application of this technique within medical imaging is Krawczyk et al. (2016), which was used to predict the degree of breast cancer. This paper deals with the imbalanced classification problem, where highest malignancy grade are less common, however, they are the most important to detect. In order to increase their detection, the authors first use an evolutionary sampling to select the most relevant samples to use as a training set by considering all instances from minority classes (high malignancy grade) and selecting instances from majority classes (low malignancy grade). Afterwards, the base classifiers are trained on the new constructed training set using the boosting technique. Combining both evolutionary sampling and boosting helped overcoming the imbalance problem and increased the detection sensitivity of highest malignancy grade cancer.

After constructing a diverse ensemble using bagging or boosting techniques, the prediction of a given input sample could be acquired by combining the outputs of all the ensemble members of the same sample. The commonly used techniques are the mean, the median, the weighted average, and voting.

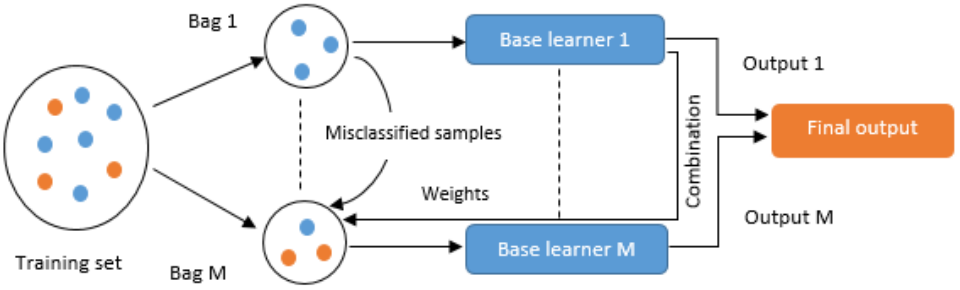


FIGURE 7.2: The boosting technique.

7.1.2.3 Stacking (Stacked Generalization)

This technique was introduced by Wolpert (1992) to increase the generalization power of the base learners. In this technique, the classifiers are trained at different levels, in other words "stacked". The dataset is first split into two partitions. After that, the base learners are trained on one of the partitions, and then used to predict the outputs of the other partition. These outputs are used as an input to train a meta learner, the role of which is to decide how to combine the outputs predicted by the previous level learners instead of doing that explicitly, which according to Wolpert (1992) leads to an increase in the generalization accuracy.

As an example, Kan et al. (2014) used this technique to transform nonfrontal face images to frontal ones by stacking three auto-encoders. Each auto-encoder contributes in the variation of an input image from nonfrontal to frontal, the result is then used as an input for the subsequent auto-encoder, until a frontal view of the input image is obtained.

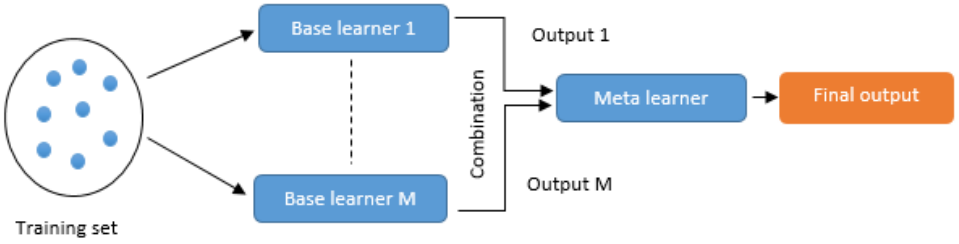


FIGURE 7.3: The stacking technique.

7.1.3 Overview of some ensemble learning algorithms

In this section, we describe two state of the art algorithms that are based on the bagging and boosting techniques.

7.1.3.1 Adaboost

Adaboost algorithm is an implementation of the boosting technique. It was introduced by Freund and Schapire (1995). The idea is to train a new learning algorithm on a new training data, which is generated based on the previous iteration: Given m training samples $(x_1, y_1), ..., (x_m, y_m)$ that belong to the same domain X, and their respective labels $y_i \in -1, +1$, a weak learner is trained on a randomly initialized distribution D_t, where $t = 1$. The training is carried out with respect to the error rate ϵ_t which should be minimized. After that, the new data distribution D_{t+1} that will be used to train the subsequent learner is computed. The process is repeated until all the weak learners are trained. The final output is the weighted combination of the weak learners outputs (Schapire, 2013).

Algorithm 3: Adaboost algorithm (Schapire, 2013).

Result: $H(x) = sign(\sum_{t=1}^{T} a_t h_t(x))$

given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in X, y_i \in -1, +1$

initialize: $D_1(i) = 1/m$ for $i = 1, ..., m$;

for t = *1,..,T* **do**

Train weak learner using distribution D_t;

Get weak hypothesis $h_t : X \rightarrow -1, +1$;

Aim: Select h_t with low weighted error:

$\epsilon_t = Pr_i \sim_{D_t} [h_t(x_i) \neq y_i]$.

Choose $a_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$

Update, for i = 1,..,m:

$D_{t+1}(i) = \frac{D_t(i) exp(-a_t y_i ht(x_i))}{Z_t}$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

end

7.1.3.2 Random Forest

Random forest was introduced by Breiman (2001). It uses the bagging algorithm to generate the training set. The ensemble in this case is composed of DTs, where all of them are trained on different bags of the original dataset. After that, each tree is built with respect to features randomness, i.e. at each node, a subset of features is randomly chosen, so that one of these features is selected as a node. The nodes of all trees in this case are randomly selected, which allows to create several "random trees", hence the name "random forest". This randomness helps

to minimise the correlation between trees by increasing the level of diversity in the ensemble. Moreover, the combination of both bagging and features randomness enhances the accuracy and improves the generalization power of the classifier. After building the ensemble, each tree votes for the predicted class, where the final output is the class having the maximum number of votes (Breiman, 2001).

7.2 Ensemble learning in the CBIR literature: an overview

Ensemble learning has not gained a large interest within the CBIR field. In addition to our contribution that is discussed in this chapter, there exists a few contributions in the literature addressing this paradigm, where the ensemble was used to empower images representations by learning diverse features (Li et al., 2017) (Huang et al., 2016) . In Li et al. (2017), a diverse ensemble was built based on varying the learning set, where multiple views were generated from each image. The motivation behind the generation of multiple views is to have a clear vision of the object, which leads to a more accurate recognition. To this end, each individual learner was trained on a different view for the purpose of extracting hash codes from it. Finally, the extracted hash codes using the ensemble were combined into a single image representation using a vote. In the second paper (Li et al., 2017), the diversity was achieved through the use of different CNN architectures to extract features from images. Similarly to Huang et al. (2016), features were combined using a majority voting scheme.

7.3 Our proposed approach

We aim to design a CBIR system based on deep learning neural networks. We propose to build an ensemble of CNNs, so that they collaborate to identify the most relevant images in the database for a query image. To this end, the CNNs to be combined must have an output layer with one output neuron for each object class in the training image database. They may have the same or different network architectures and parameter settings. Let us denote D the number of object classes. Before the ensemble is built, each CNN is trained on the images of the training database. One hot encoding is to be employed, i.e. the desired output associated to each image is a unit vector of size D with a one at the vector component corresponding to the correct class, and zeros elsewhere. Under this setup, each CNN is expected to approximate the probabilities that the input object belongs to each of the D classes under consideration:

$$f(\mathbf{X}) = (P(C_1|\mathbf{X}), ..., P(C_D|\mathbf{X})) \in [0,1]^D \tag{7.1}$$

where $\mathbf{X}$ is the input image, and C_i is the i-th class, with $i \in \{1, ..., D\}$.

After all the CNNs are trained under the above specifications, an ensemble can be built from them. Let us denote M the number of CNNs to be combined, i.e. the ensemble size. Then the class probabilities can be estimated by the ensemble by averaging the estimations coming from each ensemble member:

$$f_{Mean}(\mathbf{X}) = \frac{1}{M}\sum_{j=1}^{M} f_j(\mathbf{X}) \in [0,1]^D \tag{7.2}$$

where $f_j(\mathbf{X})$ is the output vector for the j-th CNN, given the input image $\mathbf{X}$. Another possibility is to use the component wise median, and then renormalize to a probability vector:

$$\hat{f}(\mathbf{X}) = \text{median}(\{f_j(\mathbf{X}) \mid j \in \{1, ..., M\}\}) \tag{7.3}$$

$$f_{Median}(\mathbf{X}) = \frac{1}{\left\|\hat{f}(\mathbf{X})\right\|_1}\hat{f}(\mathbf{X}) \tag{7.4}$$

where $\|\cdot\|_1$ stands for the L1-norm (Manhattan norm) of a vector.

After the ensemble has been built, it can be employed to retrieve the most similar images in the database, given a user query. Let us denote $\mathbf{X}_{Query}$ the query image, and $\mathbf{X}_j$ the training database images, where $j \in \{1, ..., N\}$ and N is the number of images in the database. We propose to rank the images in the database according to the p-norm distances between the ensemble probability vector $f_h(\mathbf{X}_{Query})$ which is obtained from the ensemble as it processes the query image, where $h \in \{Mean, Median\}$ is the ensemble type, and the probability vectors $f_h(\mathbf{X}_j)$ that are yielded by the ensemble as it processes the database images. Given these considerations, the image in the database with the best similarity to the query can be computed as follows:

$$s = \arg\min_{j \in \{1,...,N\}} \|f_h(\mathbf{X}_{Query}) - f_h(\mathbf{X}_j)\|_p \tag{7.5}$$

where $\|\cdot\|_p$ stands for the p-norm of a vector. Here p is a parameter which must be optimized by experimentation.

If we are interested in the k most similar database images given a query image, a ranking list can be obtained:

$$R_Q = \{\mathbf{X}_1^*, \mathbf{X}_2^*, ..., \mathbf{X}_k^*\} \tag{7.6}$$

R_Q is a subset of the training image database $\{\mathbf{X}_j : j = 1, ..., N\}$ where $\mathbf{X}_i^*$ is ranked before $\mathbf{X}_j^*$ when

$$\|f_h(\mathbf{X}_{Query}) - f_h(\mathbf{X}_i^*)\|_p < \|f_h(\mathbf{X}_{Query}) - f_h(\mathbf{X}_j^*)\|_p$$

Such a ranking process can be used to evaluate the performance of our CBIR system or when a parameter must be tuned.

If the number of results is not specified, then a similarity threshold ν should be used. In this case, the database images $\mathbf{X}_j \in R_Q$ are regarded as close to the input image if they satisfy the following condition :

$$\|f_h(\mathbf{X}_{Query}) - f_h(\mathbf{X}_j)\|_p < \nu \tag{7.7}$$

It is possible to add a simple way to refine the search of the closest images to a given query image using Average Query Expansion on the model proposed above. The proposed method is inspired by KNORA-UNION algorithm for dynamic ensemble selection Ko et al. (2008). The basic idea is described below.

Given an ensemble of M CNNs, and a query image $\mathbf{X}_{Query}$, after retrieving the most relevant images for this query (using our proposal: mean or median ensemble) we obtain $R_Q = \{\mathbf{X}_1^*, \mathbf{X}_2^*, ..., \mathbf{X}_k^*\}$.

We consider that a database image $\mathbf{X}_j^* \in R_Q$ which belongs to the class m is correctly classified by a CNN i in the ensemble if:

$$\max_{i=1,...,D} \left\{P^i(C_i|\mathbf{X}_j^*)\right\} = P^i(C_m|\mathbf{X}_j^*) \tag{7.8}$$

Let $\mathbf{V}_{ij}$ be the probability vector corresponding to the output of CNN_i that correctly classifies the image $\mathbf{X}_j^* \in R_Q$, that is:

$$\mathbf{V}_{ij} = (P^i(C_1|\mathbf{X}_j^*), ..., P^i(C_D|\mathbf{X}_j^*)) = f_i(X_j^*) \tag{7.9}$$

In that situation we generate a new query image vector by taking the average of the original $f_h(\mathbf{X}_{query})$ and the selected vectors in the following set:

$$\{\mathbf{V}_{ij} : i \in S_1 \subset \{1, ..., M\}, j \in S_2 \subset \{1, 2, ..., k\}\} \tag{7.10}$$

Thus, the suppression of misclassified samples allows the controlled construction of extended queries.

The designed CBIR system is described in Fig. 7.4, where a flowchart allows to follow the process from training the model to testing it. In the training phase, if the used CNNs share the same architecture, diversity is achieved through splitting the training images into M bags, which are used to train M classifiers. However, if the CNNs have different architectures, each CNN is trained on the whole training set, since the variation of the architecture is supposed to make a diverse ensemble. After that, the extracted class probability vectors from the training images are combined and stored in the database. In the testing phase, the trained classifiers are used to extract the query class probability vectors. Similarly to the training images vectors, the query vectors are combined to generate a single query representation. The p-norm distance is then used for similarity measurement in order to retrieve the k most similar images. Optionally, we can enhance the returned list of similar images using Average Query Expansion. As explained above, the base vectors that belong to the ranked list of images are selected and combined with the query vector. Finally, the retrieval process is carried out again.

7.4 Experiments and results

In this section, we give the material and methods used, as well as the conducted experiments and their respective results.

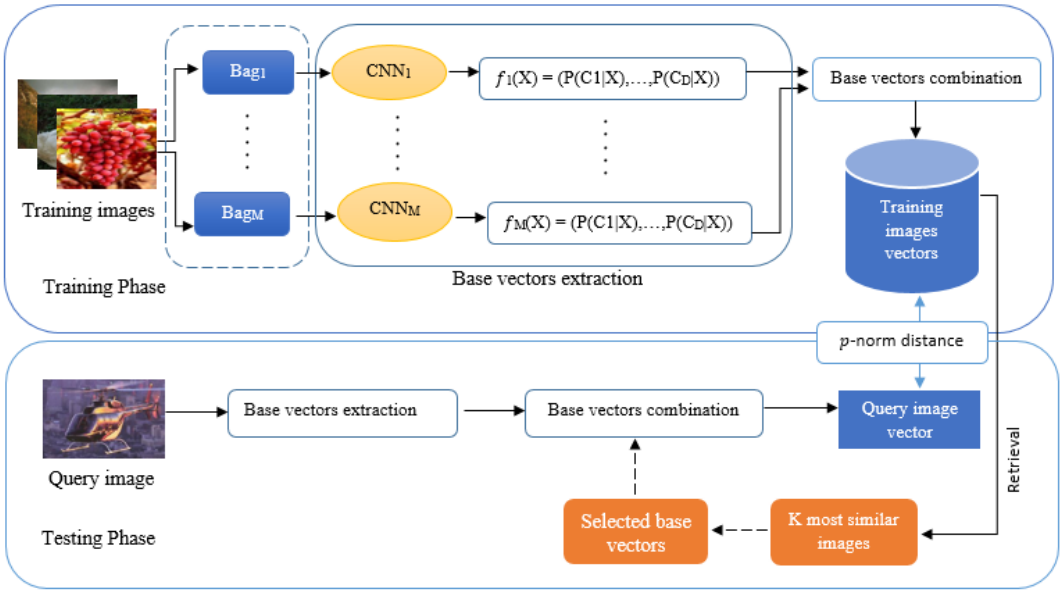


FIGURE 7.4: Schema of our proposal. The dashed frame corresponds to the bagging technique which is carried out only if the ensemble is composed of the same CNN architecture. The dashed arrows correspond to the Average Query Expansion process which can be applied on the query one time.

7.4.1 Methods

We have considered several recent well-known methods from the literature in order to compare our proposal with them. We took into consideration the best results of each competitor. The first method that we have selected is the approach that we denote as *Wan* (Wan et al., 2014b). In this work, images features are extracted from the last three fully connected layers and refined by similarity learning using the Online Algorithm for Scalable Image Similarity Learning ("OASIS").

We also consider a method that we denote *Cai* (Cai et al., 2017). In this algorithm, a triplet CNN-based architecture is used for feature extraction. First, the triplet CNN is retrained on the used dataset. After that, the features are extracted from the last two fully connected layers and then combined in order to obtain an efficient representation with different semantic levels.

The last competitor method is denoted as *Fu* (Fu et al., 2016c). It uses a CNN to extract images features. After that, a SVM is trained on the dataset to differentiate between similar images and dissimilar ones.

Regarding our proposal, it uses several CNNs having the same or different architecture. This way, well-known CNNs have been employed in the experiments:

- *VGG* Griffin et al. (2007): This architecture is characterized by the small size of convolutional filters receptive field (3x3). Some of the convolutional layers are followed by Max-pooling layers, which perform the spatial pooling. The used variants of this architecture are: VGG16 and VGG19. They consist of 16 and 19 weights layers, respectively, hence their names.

- *ResNet* He et al. (2016a): The main characteristic of this network is its depth which is up to 152 layers. Moreover, it has residual connections that decrease the complexity of the network and make its training easier. Similar to VGG, this architecture has 3x3 filter size for convolutional layers. We use several variants of this architecture depending on the network depth: ResNet50 (50 layers), ResNet101 (101 layers), and ResNet152 (152 layers).

- *ResNetV2* He et al. (2016b): ResNetV2 is an improved version of ResNet. The main difference is in use of identity skip connections and identity after-addition activation to fasten the propagation of information. Similar to ResNet, this architecture is available with different depths, resulting in three variants: Resnet50V2 (50 layers), ResNet101V2 (101 layers), and ResNet152V2 (152 layers).

- *Inception* Szegedy et al. (2015): The variants that we use are InceptionV3Szegedy et al. (2016) and InceptionResNet Szegedy et al. (2017). The latter are successors of InceptionV1 (GoogleNet) and they inherit the use of special layers called "Inception layers/modules". Instead of doing a single operation on an input, an inception module allows several operations in parallel: 1x1 convolution, 1x1 convolution followed by 3x3 convolution, 1x1 convolution followed by 5x5 convolution, and max-pooling followed by 1x1 convolution. The outputs are then concatenated and passed through the network. Moreover, the use of convolutions of different filter sizes allows capturing patterns at different scales. While, InceptionV3 reduces the filter size to speed up the training, InceptionResNet benefits from both ResNet and InceptionV1 architectures by using residual inception layers, which reduces the architecture's complexity.

- *Xception* Chollet (2017): It is worth mentioning that Xception is inspired by Inception architecture, i.e. it allows using several operations in the same layer. However, unlike Inception these operations are restricted to convolutions. Thus, Inception modules are replaced by special convolutions called "depthwise separable convolutions", where, instead

of having one convolution, several independent convolutions followed by a pointwise convolution are carried out. Note that a pointwise convolution has a filter size of 1x1. The replacement of Inception modules by depthwise separable convolutions led to a decrease in the computational cost. Moreover, this architecture defeats InceptionV3 on the ImageNet dataset in terms of accuracy.

- *MobileNet* Howard et al. (2017): We use the two versions of MobileNet architecture: MobileNet and MobileNetV2. This architecture is designed for mobiles and embedded vision applications. Similar to Xception, it uses depthwise separable convolutions to reduce the computational cost. However, the filter size is fixed to 3x3. All layers (except the fully connected layer) are followed by a batch normalization and a *relu* non linearity activation. In addition, MobilenetV2 Sandler et al. (2018) improves this architecture by the use of an inverted residual structure, and the suppression of the non linearity, which enhances the accuracy.

- *Densenet* Huang et al. (2017): The concept of Densenet architecture is based on residual connections introduced in ResNet. The idea is to connect each layer to all subsequent layers, in a manner that every layer has the outputs of all predecessor layers. This leads to a better propagation of information and speeds up the training. We use this architecture with different depths: DenseNet-121 (121 layers), DenseNet-169 (169 layers), and DensNet-201 (201 layers).

- *NasNet* Zoph et al. (2018): The main novelty is that the architecture is learnt on a small dataset (Cifar Krizhevsky et al. (2009)) using reinforcement learning, after that this architecture is used to train the network on a large scale dataset (ImageNet). Moreover, it uses "ScheduledDropPath" as a new regularization technique, which improves the model's accuracy. The concept of the variants NasNetMobile and NasNetLarge is the same, with the difference that NasNetMobile is conceived for mobile devices.

In order to code our approach, the used programming language was Python with its machine learning library Keras[1] running on top of Tensorflow[2].

The experiments have been conducted on Google colab[3] running on a NVIDIA Tesla k80 GPU.

[1] `https://keras.io/`
[2] `https://tensorflow.org/`
[3] `https://colab.research.google.com/`

7.4.2 Datasets

Our proposal is evaluated on two benchmark datasets: Caltech256 and ImageNet datasets (see Chapter 2).

In regards to Caltech256 dataset, for a fair comparison with methods Wan Wan et al. (2014b), Cai Cai et al. (2017) and Fu Fu et al. (2016c), we follow the same experimental setting by using subsets of 20 and 50 classes from Caltech256 dataset, and images from each class were randomly split into training and testing sets of 40 and 25 images, respectively. Moreover, 15 images from each category were randomly selected for validation. In each subset, the classes were selected as suggested in Chechik et al. (2010). The motivation behind this selection is to include diverse categories in terms of their semantics and their classification difficulty as measured by Griffin et al. (2007), where it is shown that the performance varies depending on the selected class.

For ImageNet dataset, the validation set is split into testing set and retrieval database. We select one query per category to evaluate our proposal, that is 1000 testing images. Moreover, we use 5000 images as a retrieval database, equally grouped into the 1000 categories. Images in both testing and validation sets were randomly selected.

7.4.3 Parameter selection

In order to achieve diversity, we evaluated two ensemble techniques to generate the pool of classifiers. The first one is the bagging technique Quinlan et al. (1996). In the corresponding experiments, M training different sets of equal size (60% from the training data) were randomly drawn with replacement, where $M = 20$. After that, each classifier was trained on a different bag of images with aim to generate an ensemble of M classifiers. This technique is evaluated on Caltech256 dataset. This way, several networks with the same architecture but different training data are considered in this first technique. The deep convolutional neural network architecture that we have employed using this technique is VGG16 Simonyan and Zisserman (2014). We have employed a fine-tuning process, where we have removed the fully connected layer from the original network and added 4 layers: A flatten layer, a dense layer with 256 neurons and a *relu* activation function, a dropout layer with a coefficient of 0.5 to prevent overfitting, and a last dense layer to specify the number of classes where the used activation function was *softmax*. The fine-tuning process was performed by training the added layers on the experimented dataset. The experimental fine tuning setting consists of a batch size equals to 64, while the loss function

is the categorical cross entropy. Additionally, the optimizer is the stochastic gradient descendant (SGD) with a learning rate of 10^{-4} and a momentum of 0.9.

The second ensemble technique is the use of classifiers having different architectures. All available pretrained CNNs in Keras library are used in this second technique, that is 18 different architectures. This second technique is evaluated on ImageNet dataset. Herein, no training phase is carried out, since the used CNNs are already trained on this dataset.

In the rest of the paper, we denote both techniques with the name of the used dataset in each experiment. That is, Caltech256 (or ensemble with networks with the same architecture) and ImageNet (or ensemble with networks with different architectures).

7.4.4 Results

In order to evaluate our proposal from a quantitative point of view, we have employed some well known performance evaluation measures. These measures are: Precision at k (P@k), Recall at k (R@k), where k is the number of images to retrieve, Precision given a distance threshold (P_ν), Recall given a distance threshold (R_ν), where ν is the threshold value, and Mean Average Precision (mAP). They provide a real number between 0 and 1, where higher is better, and they are given in the following equations:

$$P@k, P_\nu = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ retrieved\ images} \tag{7.11}$$

$$R@k, R_\nu = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ relevant\ images} \tag{7.12}$$

$$mAP = \frac{1}{L}\sum_{l=1}^{L} AP_l \tag{7.13}$$

where AP_l is the average precision for a query l, and L is the total number of queries.

Additionally, the average precision is defined by:

$$AP_l = \sum_{k=1}^{K} P_l@k \times (R_l@k - R_l@(k-1)) \tag{7.14}$$

Please note that the number of the required iterations in equation (7.14) is the number of the retrieved images k. Generally, K is the number of the required iterations to achieve a perfect

recall. In our proposal, setting K to that value for all the conducted experiments is very time consuming. Thus, we set K to the number of relevant images given a query, which is 40 for Caltech256 dataset, and 5 for ImageNet dataset.

If the number of images to retrieve is defined, precision and recall are denoted as $P@k$ and $R@k$, respectively. If a distance threshold is given, then the two measures are referred to as P_ν for precision, and R_ν for recall.

We have studied the impact of several parameters on the performance of our proposal. The first parameter p is used to compute the p-norm distance. In order to improve the retrieval results, several p-norm distances were evaluated in the retrieval step. Table 7.1 shows a comparison between the p-norm distances for the considered p values on both Caltech256 and ImageNet datasets. Depending on the parameter p, the considered distances are: Manhattan distance ($p = 1$), 1.5-norm distance ($p = 1.5$), Euclidean distance ($p = 2$), and Tchebychev distance ($p = \infty$). Results show that Manhattan distance outperformed other tested p-norm distances using both ensemble methods. Based on this, in all subsequent experiments Manhattan distance is considered to measure the similarity between the queries and the database images.

P	Mean	Median
1	**0.7363**	**0.744**
1.5	0.7275	0.7206
2	0.7209	0.7194
∞	0.7079	0.7417

(A) Caltech256 20 classes

P	Mean	Median
1	**0.5726**	**0.558**
1.5	0.5597	0.5418
2	0.5514	0.5402
∞	0.5419	0.5526

(B) Caltech256 50 classes

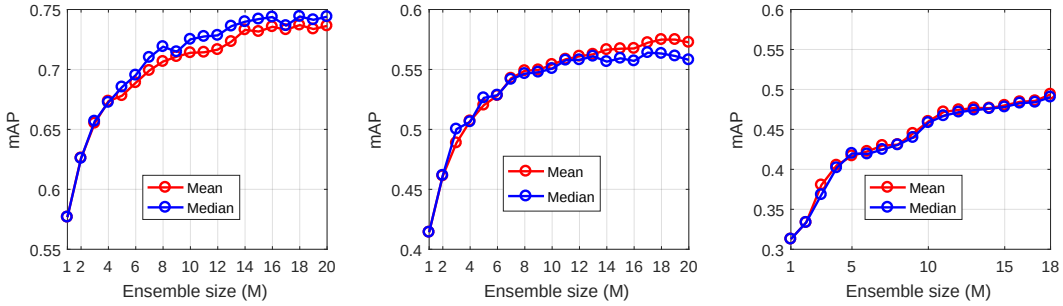
P	Mean	Median
1	**0.4939**	**0.4904**
1.5	0.4593	0.4657
2	0.4549	0.4642
∞	0.4474	0.4649

(C) ImageNet

TABLE 7.1: A comparison between several p-norm distances in terms of mAP on Caltech256 and ImageNet datasets, where $M = 20$ for Caltech256, and $M = 18$ for ImageNet. Both mean and median are considered in the comparison, best results are highlighted in **bold**.

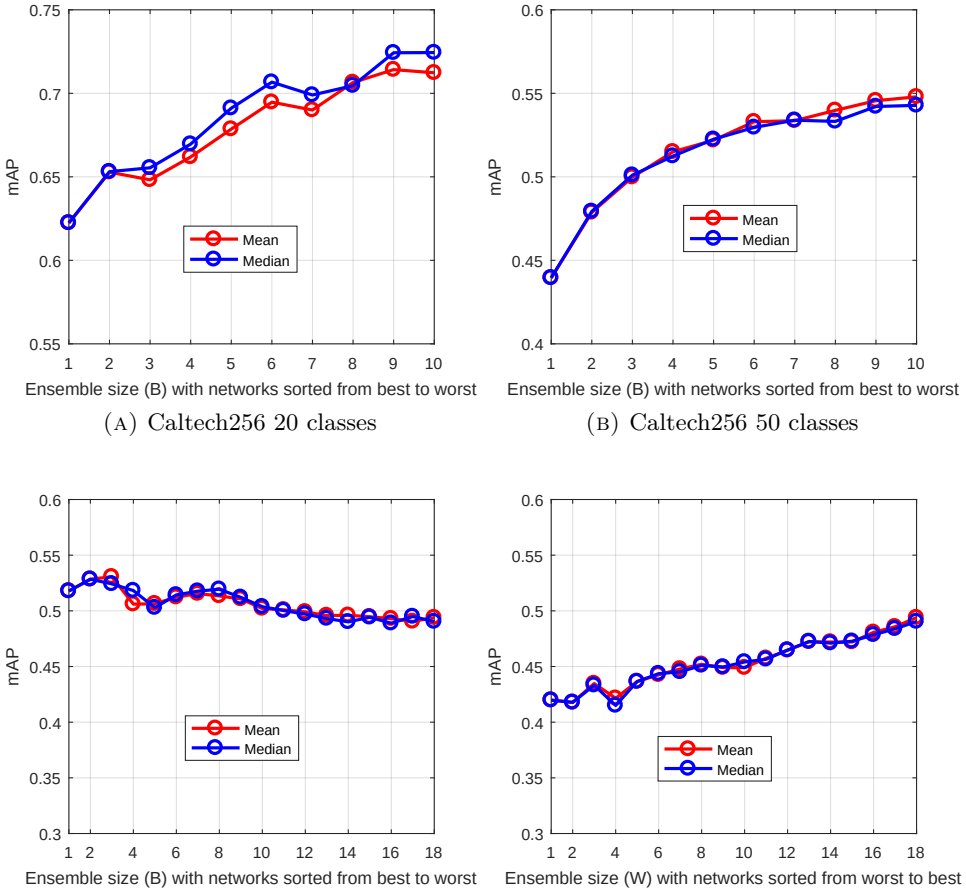
Moreover, we have studied the impact of the ensemble size M on the mAP. The results are shown in Fig. 7.5. For Caltech256 dataset, we can observe a positive correlation between the mAP and the parameter M, where for 20 classes $1 \leq M \leq 14$ using the mean, and $1 \leq M \leq 8$ using the median. For 50 classes, $1 \leq M \leq 18$ using the mean, and $1 \leq M \leq 13$ using the median. The highest mAP for 20 classes is 0.7370 using the mean, and 0.7442 using the median. The corresponding ensemble size is $M = 18$ for both methods.

For 50 classes, The highest mAP is 0.5749 using the mean, where $M = 18$. Using the median, the most accurate ensemble is of size $M = 17$, the corresponding mAP is 0.5640.



(A) Caltech256 20 classes (B) Caltech256 50 classes (C) ImageNet

FIGURE 7.5: Impact of the ensemble size on the mAP for Caltech256 and ImageNet datasets where the considered methods are mean and median.



(A) Caltech256 20 classes (B) Caltech256 50 classes

(C) ImageNet B-W (D) ImageNet W-B

FIGURE 7.6: Impact of the ensemble size with respect to the classifiers accuracy on the mAP for Caltech256 and ImageNet datasets, where the considered methods are mean and median.

For the same dataset, the most important margin in terms of mAP was recorded between ensemble sizes M=1 and M=2, where the mAP increased significantly. We refer the reason for this to the evolution of the architecture from individual CNN-based to CNN-based ensemble.

Moreover, we can observe that the positive correlation is more significant for $1 \leq M \leq 10$. Outside this interval, the increase in performance is less important. Thus, we stop the evolution of the ensemble at $M = 20$.

Based on these results, the ensemble size M is set to the value that corresponds to the highest mAP in all the following experiments. That is, for 20 classes $M = 18$ using the mean and the median, for 50 classes $M = 18$ using the mean, and $M = 17$ using the median.

The general decrease in the classification performance when we pass from 20 classes to 50 classes is due to the greater difficulty in separating more classes. Our ensemble proposal is not related to this performance decrease, since it can be seen in Figure 7.5 that the effect is present even for just a single CNN (ensemble size $M = 1$), i.e. no ensemble. Moreover, according to the study presented in Griffin et al. (2007), a negative correlation is shown between the number of classes and the measured performance, where a significant decrease in performance was recorded between 20 classes and 50 classes.

Regarding ImageNet dataset, a significant improvement is observed for $1 \leq M \leq 5$ using the mean and the median. For $5 \leq M \leq 18$, the correlation between the ensemble size M and the mAP is still positive. However, the improvement is less important. Note that the highest mAP was reached with $M = 18$ for both mean and median, where mAP = 0.4939 for the mean, and mAP = 0.4904 for the median. Even though the accuracy is generally increased after increasing the ensemble size M, the ensemble was outperformed by the single best classifier (NasNetLarge) whose mAP was 0.5178. Thus, we investigated two other ways for ensemble construction. In the first one, the ensemble was built with respect to the classifiers accuracy. B is set to the number of the most accurate classifiers. the B classifiers were sorted based on their top-5 accuracy on the ImageNet validation set, from best to worst (B-W). After that, several ensembles of size B were created, where B is set to the number of the most accurate classifiers. Moreover, $1 \leq B \leq M$. So that, $B = 1$ corresponds to the most accurate classifier, $B = 2$ corresponds to the two most accurate classifiers ensemble and so on. These ensembles are evaluated in terms of mAP as shown in Figure 7.6. This way, the ensemble could achieve a better accuracy and outperformed the single best classifier. The highest mAP reached by the mean was 0.5306, which corresponds to $B = 3$. In addition, the highest mAP reached by the median was 0.5284, where $B = 2$.

In the same way, we ordered the CNNs from worst to best (W-B), and created ensembles of size W. Results can be observed in Figure 7.6. Generally, there is an improvement in the mAP as

the ensemble size W increases. The highest mAP value was reached with W=18 using the mean and the median, where it equals 0.4939 for the mean, and 0.4904 for the median. It should be highlighted how the ensemble B-W performance decreases when networks are added; on the other hand, the ensemble W-B performance is higher when networks are added. As it can be observed, the performance of the ensemble not only depends on the accuracy of the network which has been added to the ensemble; it also depends on the networks which belong to it.

Based on these results, for ImageNet dataset M is set to B in all subsequent experiments.

We repeated the experiment where classifiers are sorted from Best to worst on Caltech256 dataset. The classifiers are sorted based on their accuracy on the training and validation sets. We selected M classifiers from the pool, where $M = 10$. As shown in Figure 7.6, in most cases, there is a positive correlation between the ensemble of the most accurate classifiers B and the mAP, where even though a less performing classifier is added to the ensemble, the larger-sized ensemble was more accurate. However, for some cases, there was a slight degradation in the performance after increasing the ensemble size B, which is possibly due to the inclusion of a less accurate classifier in the pool. Note that the highest mAP for 20 classes was reached with $B = 9$ using the mean, where it equals 0.7142, and $B = 10$ using the median, where it equals 0.7244. For 50 classes, $B = 10$ corresponds to the best performing ensemble with 0.5428 for the mean and 0.5479 for the median.

The retrieval of similar images can be done either by defining the number of images to retrieve, or by considering a threshold for the distance between the query and the database images. After deciding the ensemble size M, we evaluated the distance threshold ν, which is another important parameter that affects the retrieval quality. In order to enhance the relevance of the returned ranked list, we have experimented 20 threshold values, where $0.1 \leq \nu \leq 2$. The impact of the value ν on both precision and recall is shown in Fig. 7.7. The threshold ν was set to 1 for Caltech256 and ImageNet datasets, which is a good compromise between precision and recall.

Taking into account these analyzed configurations, the performance of our CNN-based methods is shown in Table 7.2. We refer to the ensemble methods as Mean and Median, where the mean and the median were used to combine the extracted class probability vectors using each ensemble member, respectively. We refer to the base classifier as base. This classifier was trained on the whole training set and used individually to extract the class probability vectors from images. For the two datasets, the results show that both ensemble methods outperform significantly the base classifier in terms of all performance evaluation measures. The exception occurs in R_ν

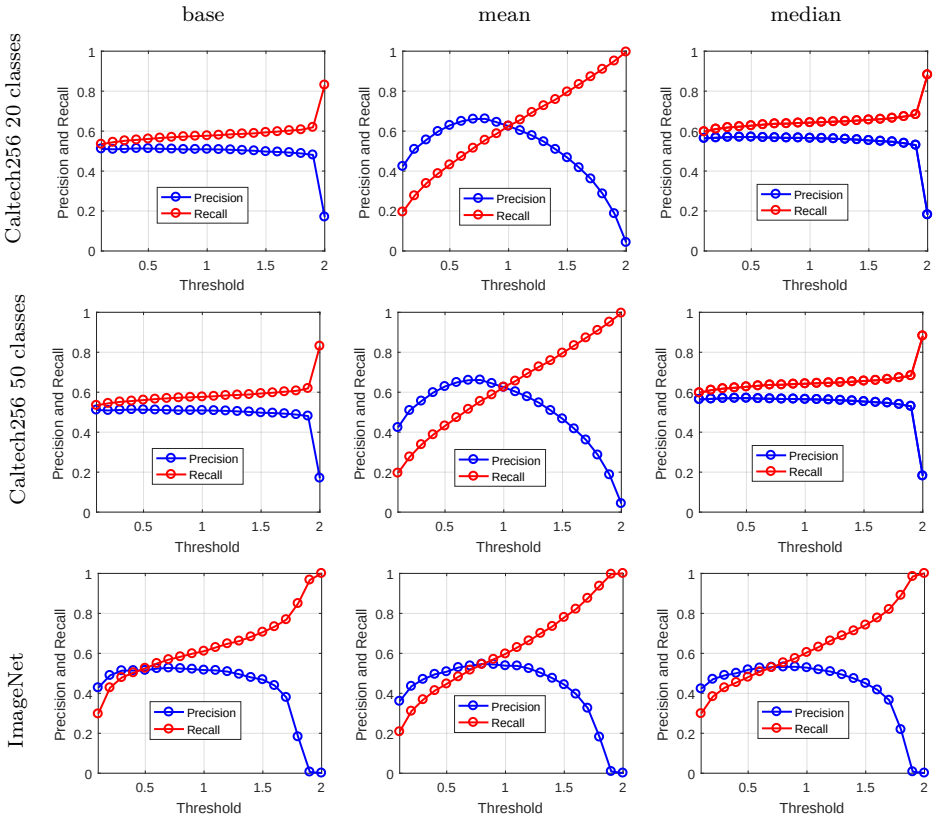


FIGURE 7.7: Precision and recall in terms of distance threshold on Caltech256 and ImageNet datasets. The considered methods are: Base, Mean, and Median, where results are shown in the left, middle, and the right of the figure, respectively. Images from the first row correspond to Caltech256 with 20 classes, images from the second row correspond to Caltech256 with 50 classes, and images from the third row correspond to ImageNet dataset.

computed on ImageNet dataset, which does not necessarily reflect the outperformance of the base in terms of this performance evaluation measure, since it depends on the used threshold value ν.

The reason behind the outperformance of the ensemble architecture is that the latter takes advantage of the lack of correlation between the base learners. This lack of correlation is produced by training the CNNs on different bags of images. Thus, the CNNs are able to learn different patterns from images and make independent errors. For instance, if a CNN misclassifies a given input, then the collaboration of other CNNs may correct this error by combining their provided outputs for this image. Consequently, the ensemble is able to provide more powerful predictions, resulting in the improvement of the retrieval quality.

Finally, we have applied the previously mentioned post processing technique called Average Query Expansion. We investigated the impact of the parameter k on the mAP. This parameter represents the number of retrieved neighbours used to generate the new query vector. Evaluated

Method	P@1	P@10	P@50	P@100	P_ν	R@1	R@10	R@50	R@100	R_ν	mAP
Base	0.692	0.6594	0.4925	0.2921	0.5401	0.0173	0.1649	0.6156	0.7302	0.5725	0.5265
Mean	**0.848**	**0.8398**	**0.6596**	**0.3613**	**0.7753**	**0.0212**	**0.2099**	**0.8246**	**0.9032**	0.7414	0.7370
Median	0.844	0.8296	0.6455	0.3554	0.7341	**0.0212**	0.2074	0.8068	0.8885	**0.7609**	**0.7442**

(A) Caltech256 20 classes

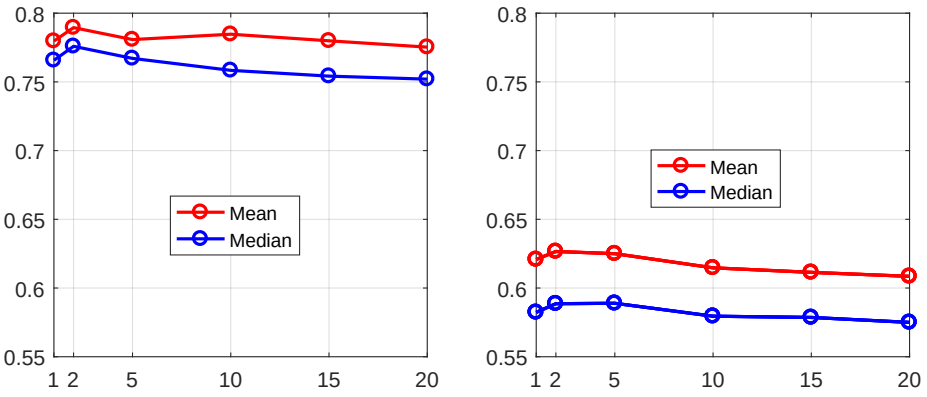
Method	P@1	P@10	P@50	P@100	P_ν	R@1	R@10	R@50	R@100	R_ν	mAP
Base	0.6408	0.6269	0.4685	0.2762	0.509	0.016	0.1567	0.5856	0.6905	0.5767	0.4909
Mean	**0.7456**	**0.7201**	**0.5483**	**0.3173**	**0.6248**	**0.0186**	**0.1800**	**0.6853**	**0.7932**	0.6258	**0.5749**
Median	0.728	0.6958	0.5198	0.2994	0.5646	0.0182	0.174	0.6498	0.7485	**0.6425**	0.5640

(B) Caltech256 50 classes

Method	P@1	P@10	P@50	P@100	P_ν	R@1	R@10	R@50	R@100	R_ν	mAP
Base	0.5970	0.3277	0.0761	0.0397	0.5161	0.1194	0.6554	0.7612	0.7950	**0.6108**	0.5178
Mean	**0.6280**	**0.3352**	**0.0791**	**0.0419**	**0.5381**	**0.1256**	**0.6704**	**0.7914**	**0.8390**	0.5972	**0.5306**
Median	0.6200	0.3332	0.0781	0.0412	0.5282	0.1240	0.6664	0.7814	0.8244	0.6046	0.5284

(C) ImageNet

TABLE 7.2: Image retrieval performance on the experimented datasets, best results are highlighted in **bold**.



(A) Caltech256 20 classes (B) Caltech256 50 classes

FIGURE 7.8: Impact of the number of neighbours considered in the Average Query Expansion technique on the mAP for Caltech256 dataset, where the considered methods are mean and median.

k values were: 1, 2, 5, 10, 15, and 20. Fig. 7.8 shows the mAP in terms of k. For 20 classes, the highest mAP was reached with $k = 2$ using the mean and the median. For 50 classes, it was reached with $k = 2$ using the mean and $k = 5$ using the median. After applying the Average Query Expansion technique to our architecture, both ensemble methods achieved better results in terms of mAP, as shown in Table 7.3.

Results can be observed from a qualitative point of view. Fig. 7.9 shows an example of a query and the top 10 retrieved images. In the given example, for Caltech256, both mean and median performed equally in terms of precision, where (p@10 = 0.9). For ImageNet dataset, the same

Method	mAP
Mean	0.7370
Mean + QE	**0.7895**
Median	0.7442
Median + QE	0.7759

(A) 20 classes

Method	mAP
Mean	0.5749
Mean + QE	**0.6266**
Median	0.5640
Median + QE	0.5889

(B) 50 classes

TABLE 7.3: mAP using the mean and the median with and without applying the Average Query Expansion to Caltech256 dataset, best results are highlighted in **bold**.

precision value was obtained by the mean and the median, where ($p@10 = 0.3$). Note that the Average Query Expansion is not applied in this retrieval example.

(a) Query

(a) Mean

(b) Median

(A) Caltech256 dataset

(a) Query

(a) Mean

(b) Median

(B) ImageNet dataset

FIGURE 7.9: A query image and the top 10 returned images from both Caltech256 and ImageNet datasets using the mean and the median.

In order to further check the performance of our proposal, we compared it against relevant retrieval methods. The comparison is carried out on Caltech256 dataset in terms of mAP. In order to be comparable with the selected methods, the parameter k defined in equation 7.14 was set to the number of iterations required to achieve a perfect recall. In addition, no post processing technique is applied to our method. Results shown in Table 7.4 confirm the efficiency of our method where both mean and median outperform the competing methods. We can also observe that the mean achieved the best mAP with a slight difference compared to the median on 20 classes. However, when evaluated on 50 classes, this outperformance was more significant.

Method	*mAP*
Wan Wan et al. (2014b)	0.7388
Cai Cai et al. (2017)	0.6726
Fu Fu et al. (2016c)	0.6106
Mean	**0.8158**
Median	0.8126

(A) 20 classes

Method	*mAP*
Wan (Wan et al., 2014b)	0.5410
Cai (Cai et al., 2017)	0.5532
Fu (Fu et al., 2016c)	0.4264
Mean	**0.6713**
Median	0.6456

(B) 50 classes

TABLE 7.4: Comparison between our proposal (mean and median) and related methods in terms of mAP, best results are highlighted in **bold**. Results indicated in percentage in original papers are converted to decimal fractions.

7.5 Conclusion

We could demonstrate the effectiveness of the ensemble learning paradigm within CBIR by proposing a novel architecture for combining the CNNs members. Moreover, we tracked the impact of several parameters on the performance of this proposed architecture and chose the best parameter settings to maximize the performance of the designed system.

The drawback of ensemble methods is their computational complexity. Not to speak of the training phase, since the latter is done offline, but the problem is especially in the inference phase where a new query is input to extract its features. The required time to predict the output of a given query depends on the ensemble size, meaning that the larger the ensemble, the more time is required for the prediction. It should be highlighted though that this has an impact only on systems where the response time is critical. Otherwise, we privilege the effectiveness over the efficiency. To sum up, the recent advance in computing devices would be of great benefit to reduce this computational issue.

General conclusions and perspectives

Conclusions

Throughout this manuscript, we attempted to emphasize the adaptation as a powerful tool in designing CBIR systems. In the first part of this thesis, we gave the basic concepts of CBIR in order to make a general overview of this area in Chapter 2. Then, in Chapter 3, we highlighted the commonly used adaptive techniques in the literature, as well as their impact on building high-performing systems. After that, we tackled the scope of this thesis which is the adaptation within CBIR in Chapter 4. Therein, we put the light on commonly used adaptive techniques in the CBIR literature, including those addressed in our study.

After clarifying the context of this thesis, the second part was dedicated to presenting our contributions in this area, as well as the main findings of our research. The first contribution entitled "Adaptive Content Based Image Retrieval based on RICE Algorithm Selection Model" was presented in Chapter 5, where we presented a novel adaptive CBIR framework based on RICE model for algorithm selection. We projected this model onto CBIR by decomposing the CBIR algorithm to many components, including feature and color space, then we used this model to select the best performing algorithm on both dataset and queries. Significant results have been obtained, revealing that this model is promising in this field. In the second contribution, we explored another technique, which is deep learning. The related work was presented in Chapter 6 entitled "Content Based Image Retrieval by Convolutional Neural Networks", where we adapted a CNN to the target dataset and used it for features extraction process. The satisfactory results were behind pursuing our research in the same direction, i.e. deep learning. In this respect, a significant extension has been made resulting in our third and main contribution, which is discussed in Chapter 7, entitled "Content Based Image Retrieval by Ensembles of Deep Learning Object Classifiers". The material extension consists in the use of larger datasets, as well as more

diverse CNNs architectures. In addition, the theoretical extension was about using an ensemble of CNNs instead of a single one in the feature extraction process. These CNNs were combined resulting in a powerful classifier, and hence a more powerful image characterization. Moreover, we tuned several parameters of the proposed architecture depending on the dataset, such as the ensemble size, which increased the retrieval quality.

Seen as a whole, the scalability could be observed along our contributions, whether in terms of the growing dataset size or the increased computational power. From the first to the second contribution, we reinforced the computational material by considering a GPU, in addition to the CPU, so as to decrease the computational complexity while training the CNN. In addition to this, in our third contribution, we worked on larger datasets in terms of size and number of classes, where we enlarged the dataset size from 1000 images grouped into 10 classes, to datasets of size up to 6000 images and 1000 classes, which also confirms the scalability of the technique proposed in this contribution.

To conclude, we had in view the exploration of some adaptive techniques in CBIR with aim to emphasize the role of adaptation in empowering CBIR systems. Thankfully, through our contributions, we could achieve the desired goal and made our small mark on the huge area of image retrieval by content.

Perspectives

In the wake of this research, some ideas have sparked our interest. These potential future works are summarized as follows:

- Extending the application of RICE model for algorithm selection in CBIR by increasing the degree of diversity between the considered CBIR-algorithms, so as to enlarge the portfolio, and/or combining the selected top k algorithms instead of using a single one, which may improve the model's performance.
- Exploring other deep learning techniques in CBIR. One possible idea could be "dynamic selection of classifiers", where we aim to choose the best performing ensemble for each query for the purpose of feature extraction.

- Finally, we show an immense interest in involving the adaptation paradigm depicted in DL techniques in other fields of study, namely in education where adaptive learning has a promising future and may revolutionize education.

Appendix A. Candidate bibliography

A. Published work and its visibility

a. Journal papers

Hamreras, S., Boucheham, B., Molina-Cabello, M. A., Benítez-Rochel, R., & López-Rubio, E. (2020). Content based image retrieval by ensembles of deep learning object classifiers. Integrated Computer-Aided Engineering, 1-15.

Paper title Content based image retrieval by ensembles of deep learning object classifiers [1]

Journal name Integrated Computer-Aided Engineering [2]

Issue title Special issue on the Interplay Between Natural and Artificial Computation (IWINAC 2019)

Publisher IOS Press [3]

Impact factor (2020) 4.709

H-index 39

Indexed in Academic Source Complete, ACM Computing Reviews, ACM Digital Library, Cambridge Scientific Abstracts, Computer Science Index, CPX, Current Mathematical Publications, DBLP Bibliography Server, EBSCO databases, Google Scholar, Inspec IET, International Abstracts in Operations Research, io-port, MasterFILE, Mathematical Reviews, MathSciNet, Microsoft Academic Search, Scopus, Ulrich's Periodicals, VINITI,

[1] https://content.iospress.com/articles/integrated-computer-aided-engineering/ica200625
[2] https://iospress.nl/journal/integrated-computer-aided-engineering
[3] https://www.iospress.nl

Web of Science: Current, Contents/Engineering, Computing and Technology, Web of Science: Journal Citation Reports/Science Edition, Web of Science: Science Citation, Index-Expanded (SciSearch®), Zentralblatt MATH.

b. International conferences

Hamreras, S., Benítez-Rochel, R., Boucheham, B., Molina-Cabello, M. A., & López-Rubio, E. (2019, June). Content Based Image Retrieval by Convolutional Neural Networks. In International Work-Conference on the Interplay Between Natural and Artificial Computation (pp. 277-286). Springer, Cham.

Paper title Content Based Image Retrieval by convolutional neural networks [4]

Conference name International Work-conference on the Interplay between Natural and Artificial Computation (IWINAC)

Digital library SpringerLink [5]

Publisher Springer [6]

Book From Bioinspired Systems and Biomedical Applications to Machine Learning [7]

Conference Location Almeria, Spain

Core ranking Yes

GII-GRIN-SCIE (GGS) Conference Rating: Work in progress

Indexed in Scopus, Google Scholar, DBLP Bibliography Server, Microsoft Academic Search

Hamreras, S., & Boucheham, B. (2018, April). Adaptive content based image retrieval based on RICE algorithm selection model. In 2018 International Symposium on Programming and Systems (ISPS) (pp. 1-6). IEEE.

[4] `https://link.springer.com/chapter/10.1007/978-3-030-19651-6_27`
[5] `https://link.springer.com`
[6] `https://springer.com`
[7] `https://link.springer.com/book/10.1007/978-3-030-19651-6`

Paper title Adaptive content based image retrieval based on RICE algorithm selection model [8]

Conference name The International Symposium on Programming and Systems (ISPS)

Digital library IEEE Xplore [9]

Publisher IEEE [10]

Conference Location Algiers, Algeria

Indexed in IEEE Xplore, Scopus, Google Scholar, Microsoft Academic Search

B. Author international visibility

ResearchGate [11] H-Index: 4.44, citations: 9

Google scholar [12] H-Index: 2, citations: 13

DBLP Computer science bibliography [13] with two indexed publications

Microsoft Academic Search [14] with three indexed publications

IEEE [15] with one indexed publication

[8] https://ieeexplore.ieee.org/abstract/document/8379022
[9] https://ieeexplore.ieee.org
[10] https://ieee.org
[11] https://researchgate.net/profile/Safa_Hamreras
[12] https://scholar.google.com/citations?user=x-TBfWEAAAAJ&hl=fr
[13] https://dblp.org/pid/240/6445.html
[14] https://academic.microsoft.com/search?q=Safa+Hamreras
[15] https://ieeexplore.ieee.org/author/37086391394

Appendix B. Cover pages of published work

Integrated Computer-Aided Engineering 27 (2020) 317–331
DOI 10.3233/ICA-200625
IOS Press

Content based image retrieval by ensembles of deep learning object classifiers

Safa Hamreras[a,*], Bachir Boucheham[a], Miguel A. Molina-Cabello[b,c], Rafaela Benítez-Rochel[b,c] and Ezequiel López-Rubio[b,c]
[a]*Laboratoire de Recherche en Électronique de Skikda and Department of Computer Science, University of Skikda 20 Août 1955, Skikda, Algeria*
[b]*Department of Computer Languages and Computer Science, University of Málaga, Málaga, Spain*
[c]*Biomedic Research Institute of Málaga, Málaga, Spain*

Abstract. Ensemble learning has demonstrated its efficiency in many computer vision tasks. In this paper, we address this paradigm within content based image retrieval (CBIR). We propose to build an ensemble of convolutional neural networks (CNNs), either by training the CNNs on different bags of images, or by using CNNs trained on the same dataset, but having different architectures. Each network is used to extract the class probability vectors from images to use them as representations. The final image representation is then generated by combining the extracted class probability vectors from the built ensemble. We show that the use of CNN ensembles is very efficient in generating a powerful image representation compared to individual CNNs. Moreover, we propose an Averarge Query Expansion technique for our proposal to enhance the retrieval results. Several experiments were conducted to extensively evaluate the application of ensemble learning in CBIR. Results in terms of precision, recall, and mean average precision show the outperformance of our proposal compared to the state of the art.

Keywords: Content based image retrieval, ensemble learning, convolutional neural networks

1. Introduction

Content Based Image Retrieval (CBIR) is the procedure of automatically identifying images by the extraction of their low-level visual features like color, texture, shape properties or any other features being derived from the image itself [1]. The well-known 'semantic-gap' issue that exists between low-level features of images and high-level semantic concepts perceived by humans has been addressed through a variety of techniques [2–6].

However, the huge diversity of semantical concepts contained in images suggest that many robust discrimination and learning techniques are needed. In that respect, deep learning has become a significant step forward in the already developing fields of computer vision. It is a technique which includes a family of machine learning algorithms that attempt to model high-level abstraction in data by employing deep architectures composed of multiple non-linear transformations [7,8]. These models are originated from computational models inspired by the structure and functions of the brain called artificial neural networks. The main reasons behind its success are the availability of large annotated datasets and the computational power and affordability of GPUs. A typical example of deep architectures are feed-forward neural networks with many hidden layers, and backpropagation [9] as a learning algorithm. Recent successes of deep learning techniques, especially Convolutional Neural Networks (CNNs) [10] in solving computer vision and image understanding tasks [11–15] have inspired us to explore this approach with aim to end up with more robust and better performing CBIR systems.

*Corresponding author: Safa Hamreras, Laboratoire de Recherche en Électronique de Skikda and Department of Computer Science, University of Skikda 20 Août 1955, Skikda, Algeria. E-mail: safahamreras@gmail.com.

ISSN 1069-2509/20/$35.00

Content Based Image Retrieval by Convolutional Neural Networks

Safa Hamreras[1], Rafaela Benítez-Rochel[2], Bachir Boucheham[1], Miguel A. Molina-Cabello[2], and Ezequiel López-Rubio[2]

[1] Department of Computer Science
University of 20 August 1955, BP 26, Route El Hadaiek, 21000 Skikda, Algeria
safahamreras@gmail.com,bachir.boucheham@hotmail.com,
[2] Department of Computer Languages and Computer Science. University of Málaga.
Bulevar Louis Pasteur, 35. 29071 Málaga. Spain.
{benitez,miguelangel,ezeqlr}@lcc.uma.es,
WWW home page: http://www.lcc.uma.es/~ezeqlr/index-en.html

Abstract. In this paper, we present a Convolutional Neural Network (CNN) for feature extraction in Content Based Image Retrieval (CBIR). The proposed CNN aims at reducing the semantic gap between low-level and high-level features. Thus, improving retrieval results. Our CNN is the result of a transfer learning technique using Alexnet pretrained network. It learns how to extract representative features from a learning database and then uses this knowledge in query feature extraction. Experimentations performed on Wang (Corel 1K) database show a significant improvement in terms of precision over the state of the art classic approaches.

Keywords: Content Based Image Retrieval, Convolutional Neural Networks, Feature extraction.

1 Introduction

The increased use of digital computers, multimedia, and storage systems over recent years has result in large image and multimedia content repositories. This huge amount of multimedia data is being used in many fields like medical treatment, satellite data, electronic games, archaeology, video and still images repository, and digital forensics and surveillance systems. That rapid growing has created an ongoing demand of retrieval images systems operating on a large scale.

Content Based Image Retrieval (CBIR) is the procedure of automatically retrieving images by the extraction of their low-level visual features, like color, texture, shape properties or any other features being derived from the image itself. The performance of a CBIR system mainly depends on these selected features [14]. Thus, it can be said that through navigation, browsing, query-by-example etc, we can calculate the similarity between the low-level image contents which can be used for the retrieval of relevant images. The most challenging issue associated with CBIR systems is reducing the semantic gap. It is the information lost

Adaptive Content Based Image Retrieval based on RICE Algorithm Selection model

Safa Hamreras and Bachir Boucheham
Department of Computer Sciences
University 20 Août 1955
Skikda, Algeria
SafaHamreras@gmail.com
Bachir.boucheham@hotmail.com

***Abstract*—In this paper, we propose a framework for "Algorithm Selection" for image retrieval by content (CBIR). The framework is based on the model of RICE and is adapted to satisfy a given query depending on its characteristics by choosing the best classical *CBIR-Algorithm* from an *Algorithm-Portfolio*. As many as six algorithms for content based image retrieval have been included in the framework as alternatives for the different queries, including the training step. These algorithms range from RGB color moments, RGB color histogram to local binary pattern (LBP), etc. Therefore, there has been put an effort in the framework to cover the basic characteristics of images: Color and texture. Also, the framework integrates two color models to better enhance the *Algorithm-Query* adaptation process. Experimentations on the Wang (Corel 1k) database show the effectiveness of the proposed framework. Indeed, enhancements of more than 4% in precision have been obtained.**

***Keywords*—Adaptive framework, Content Based Image Retrieval, Algorithm Selection problem, feature selection.**

I. INTRODUCTION

There exists two approaches in image retrieval: TBIR (Text Based Image Retrieval) and CBIR (Content Based Image Retrieval). The first one represents images by text annotations. This approach has many flaws: First, choosing the adequate annotation for a given image might be a difficult task, especially when this image contains a big quantity of information. Second, it is a manual task, an annotator is charged by giving images the suitable annotation which takes a lot of time and effort. Finally, the choice of image description is subjective, the same image may have multiple annotations depending on the person describing it [1]. One of the most interesting tentatives to overcome TBIR flaws was "Content Based Image Retrieval". Unlike text based image retrieval, this approach uses low level features to describe and retrieve images such as color, texture and shape.

Basic CBIR systems follow two phases: Offline indexation and query satisfaction. The first phase consists in extracting low level features to characterize an image. The search step includes feature extraction of the query image followed by the measure of similarity between the query image and all database images. Similarity is calculated using a distance function, and there exist many: Euclidian distance, Manhattan distance, Chebyshev distance, etc. After that, the CBIR system extracts the relevant images according to obtained similarity measures, the smaller the distance, the more similar images are. In other words, this step consists in "searching the k images whose feature vectors are most similar to the feature vector of the query image, namely k-nearest neighbor (k-NN) Searching" [2].

Advanced CBIR applications use many techniques and approaches to achieve high quality retrievals. These techniques include: Features selection, relevance feedback, Support Vector Machines (SVM), etc. In particular, features selection is one of the most interesting approaches in image retrieval.

Knowing that there is no feature that outperforms all others in the general case [3] [4], if we consider that one specific feature can be used as a unique feature in an image matching algorithm, then the feature selection problem can be regarded as an algorithm selection problem. However, a CBIR-Algorithm needs in fact at least two ingredients: Feature and color space. In this work we present a novel framework for feature selection based on RICE model for algorithm selection [5]. Our framework selects the best (Feature, Color space) to apply on a given query based on training data.

The rest of this paper is organized in 3 sections: First, we will expose the algorithm selection problem and its related concepts in section II, then we will explain the functioning of our framework in section III, and last we will report the obtained results and discuss them in section IV.

II. THE ALGORITHM SELECTION PROBLEM

The algorithm selection problem was defined by John RICE in 1976 [5]. It consists in choosing the best algorithm to apply to solve a specific problem instance. "The best algorithm" means here: The algorithm that has the best performance compared to all other algorithms, given a problem instance [6]. Researchers have recognized that there is no algorithm that has the best overall performance, but choosing the best algorithm to apply on each problem instance can lead to improve the average performance [7]. Fig. 1 illustrates RICE model for algorithm selection as presented in [5]:

978-1-5386-4690-8/18/$31.00 ©2018 IEEE

Bibliography

Acharya, T. (2002). Integrated color interpolation and color space conversion algorithm from 8-bit bayer pattern rgb color space to 24-bit cie xyz color space. US Patent 6,366,694.

Ahmad, J., Muhammad, K., Bakshi, S., and Baik, S. W. (2018). Object-oriented convolutional features for fine-grained image retrieval in large surveillance datasets. *Future Generation Computer Systems*, 81:314–330.

Ahmed, E., Jones, M., and Marks, T. K. (2015). An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3908–3916.

Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. IEEE.

Alhwarin, F., Ristić-Durrant, D., and Gräser, A. (2010). Vf-sift: very fast sift feature matching. In *Joint Pattern Recognition Symposium*, pages 222–231. Springer.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

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.

Altieri, M. A. and Nicholls, C. I. (2017). The adaptation and mitigation potential of traditional agriculture in a changing climate. *Climatic Change*, 140(1):33–45.

Andaló, F. A., Miranda, P. A., Torres, R. d. S., and Falcão, A. X. (2010). Shape feature extraction and description based on tensor scale. *Pattern recognition*, 43(1):26–36.

Andreyev, A., Sitek, A., and Celler, A. (2011). Fast image reconstruction for compton camera using stochastic origin ensemble approach. *Medical physics*, 38(1):429–438.

Andrysiak, T. and Choraś, M. (2005). Image retrieval based on hierarchical gabor filters. 15(4):471—480.

Arandjelovic, R. and Zisserman, A. (2013). All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585.

Arivazhagan, S. and Ganesan, L. (2003). Texture classification using wavelet transform. *Pattern recognition letters*, 24(9-10):1513–1521.

Babenko, A., Slesarev, A., Chigorin, A., and Lempitsky, V. (2014). Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer.

Backes, A. R., Martinez, A. S., and Bruno, O. M. (2011). Texture analysis using graphs generated by deterministic partially self-avoiding walks. *Pattern Recognition*, 44(8):1684–1689.

Banerjee, I., Kurtz, C., Devorah, A. E., Do, B., Rubin, D. L., and Beaulieu, C. F. (2018). Relevance feedback for enhancing content based image retrieval and automatic prediction of semantic image features: Application to bone tumor radiographs. *Journal of biomedical informatics*, 84:123–135.

Banerjee, S. and Hecker, J. P. (2017). A multi-agent system approach to load-balancing and resource allocation for distributed computing. In *First Complex Systems Digital Campus World E-Conference 2015*, pages 41–54. Springer.

Belattar, K., Mostefai, S., and Draa, A. (2018). A hybrid ga-lda scheme for feature selection in content-based image retrieval. *International Journal of Applied Metaheuristic Computing (IJAMC)*, 9(2):48–71.

Ben-Hur, A. and Weston, J. (2010). A user's guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer.

Benloucif, S. and 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)*, pages 1–7. IEEE.

Bicego, M., Lagorio, A., Grosso, E., and Tistarelli, M. (2006). On the use of sift features for face authentication. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 35–35. IEEE.

Bougueroua, S. and Boucheham, B. (2017). Apport de la texture à la caractérisation d'images : Application à la recherche de l'image par le contenu visuel (CBIR). Ph.D thesis. University of Skikda - 20 Août 1955.

Bougueroua, S. and Boucheham, B. (2018). Glibp: Gradual locality integration of binary patterns for scene images retrieval. *Journal of Information Processing Systems*, 14(2).

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Buchanan, B. G. and Duda, R. O. (1983). Principles of rule-based expert systems. In *Advances in computers*, volume 22, pages 163–216. Elsevier.

Buyukyilmaz, M. and Cibikdiken, A. O. (2016). Voice gender recognition using deep learning. In *2016 International Conference on Modeling, Simulation and Optimization Technologies and Applications (MSOTA2016)*. Atlantis Press.

Cai, Z., Gao, W., Yu, Z., Huang, J., and Cai, Z. (2017). Feature extraction with triplet convolutional neural network for content-based image retrieval. In *IEEE Conference on Industrial Electronics and Applications*, pages 337–342. IEEE.

Calderín, J. F., Masegosa, A. D., and Pelta, D. A. (2015). Algorithm portfolio based scheme for dynamic optimization problems. *International Journal of Computational Intelligence Systems*, 8(4):667–689.

Carpineto, C. and Romano, G. (2012). A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, 44(1):1–50.

Castelvecchi, D. (2016). Can we open the black box of ai? *Nature News*, 538(7623):20.

Chakravarti, R. and Meng, X. (2009). A study of color histogram based image retrieval. In *2009 Sixth International Conference on Information Technology: New Generations*, pages 1323–1328. IEEE.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Chang, J., Wang, L., Meng, G., Xiang, S., and Pan, C. (2017). Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887.

Chang, Y.-C., Kao, W.-Y., Chu, C.-P., and Chiu, C.-H. (2009). A learning style classification mechanism for e-learning. *Computers & Education*, 53(2):273–285.

Chechik, G., Sharma, V., Shalit, U., and Bengio, S. (2010). Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818.

Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., and Inkpen, D. (2016). Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Chen, S. and Zhang, J. (2008). The adaptive learning system based on learning style and cognitive state. In *2008 International Symposium on Knowledge Acquisition and Modeling*, pages 302–306. IEEE.

Cherrington, M., Thabtah, F., Lu, J., and Xu, Q. (2019). Feature selection: filter methods performance challenges. In *2019 International Conference on Computer and Information Sciences (ICCIS)*, pages 1–4. IEEE.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

Chum, O., Philbin, J., Sivic, J., Isard, M., and Zisserman, A. (2007). Total recall: Automatic query expansion with a generative feature model for object retrieval. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE.

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague.

Cyganek, B. (2012). One-class support vector ensembles for image segmentation and classification. *Journal of Mathematical Imaging and Vision*, 42(2-3):103–117.

Da Silva, S. F., Ribeiro, M. X., Neto, J. d. E. B., Traina-Jr, C., and Traina, A. J. (2011). Improving the ranking quality of medical image retrieval using a genetic feature selection method. *Decision support systems*, 51(4):810–820.

Daily, M., Medasani, S., Behringer, R., and Trivedi, M. (2017). Self-driving cars. *Computer*, 50(12):18–23.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(3):131–156.

Dawood, H., Alkinani, M. H., Raza, A., Dawood, H., Mehboob, R., and Shabbir, S. (2019). Correlated microstructure descriptor for image retrieval. *IEEE Access*, 7:55206–55228.

de Ves, E., Benavent, X., Coma, I., and Ayala, G. (2016). A novel dynamic multi-model relevance feedback procedure for content-based image retrieval. *Neurocomputing*, 208:99–107.

Deb, A. (2011). Introduction to soft computing techniques: artificial neural networks, fuzzy logic and genetic algorithms. In *Soft Computing in Textile Engineering*, pages 3–24. Elsevier.

Deng, L., Yu, D., et al. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387.

Díaz, F. S., Rubilar, T. P., Figueroa, C. C., and Silva, R. M. (2018). An adaptive e-learning platform with vark learning styles to support the learning of object orientation. In *2018 IEEE World Engineering Education Conference (EDUNINE)*, pages 1–6. IEEE.

Dietterich, T. G. et al. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, 2:110–125.

Drucker, H., Wu, D., and Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5):1048–1054.

Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77.

Ekta Walia, Aman Pal (2014). Fusion framework for effective color image retrieval. *J. Vis. Commun. Image R.*

Eljawad, L., Aljamaeen, R., Alsmadi, M., Almarashdeh, I., Abouelmagd, H., Alsmadi, S., Haddad, F., Alkhasawneh, R., Alazzam, M., et al. (2019). Arabic voice recognition using fuzzy logic and neural network. *International Journal of Applied Engineering Research*, 14(3):651–662.

Emerson, C. W., Siu-Ngan Lam, N., and Ouattrochi, D. (1999). Multi-scale fractal analysis of image texture and patterns. *Photogrammetric Engineering and Remote Sensing*, 65:51–62.

Eren, Y., Küçükdemiral, İ. B., and Üstoğlu, İ. (2017). Introduction to optimization. In *Optimization in Renewable Energy Systems*, pages 27–74. Elsevier.

Fachrurrozi, M. et al. (2017). Multi-object face recognition using content based image retrieval (cbir). In *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 193–197. IEEE.

Fachrurrozi, M., Saparudin, S., Erwin, E., Mardiana, M., Badillah, C. F., Erlina, J., and Lazuardi, A. (2018). Real-time multi-object face recognition using content based image retrieval (cbir). *International Journal of Electrical and Computer Engineering*, 8(5):2812.

Fan, F., Xiong, J., and Wang, G. (2020). On interpretability of artificial neural networks. *arXiv preprint arXiv:2001.02522*.

Fan, H., Zheng, L., Yan, C., and Yang, Y. (2018). Unsupervised person re-identification: Clustering and fine-tuning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(4):1–18.

Fathian, M., Tab, F. A., Moradi, K., and Saien, S. (2018). A learning automata framework based on relevance feedback for content-based image retrieval. *International Journal of Machine Learning and Cybernetics*, 9(9):1457–1472.

Feldman, R. C., Aldana, E., and Stein, K. (2019). Artificial intelligence in the health care space: How we can trust what we cannot know. *Stan. L. & Pol'y Rev.*, 30:399.

Flores-Mendez, R. A. (1999). Towards a standardization of multi-agent system framework. *XRDS: Crossroads, The ACM Magazine for Students*, 5(4):18–24.

Freund, Y. and Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer.

Fu, K., Cheng, D., Tu, Y., and Zhang, L. (2016a). Credit card fraud detection using convolutional neural networks. In *International Conference on Neural Information Processing*, pages 483–490. Springer.

Fu, R., Li, B., Gao, Y., and Wang, P. (2016b). Content-based image retrieval based on cnn and svm. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 638–642. IEEE.

Fu, R., Li, B., Gao, Y., and Wang, P. (2016c). Content-based image retrieval based on cnn and svm. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 638–642. IEEE.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., and Zettlemoyer, L. (2018). Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Gionis, A., Indyk, P., Motwani, R., et al. (1999). Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529.

Glowacka, D., Teh, Y. W., and Shawe-Taylor, J. (2016). Image retrieval with a bayesian model of relevance feedback. *arXiv preprint arXiv:1603.09522*.

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., Holmes, G., and Abdessalem, T. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9-10):1469–1495.

González-Briones, A., Villarrubia, G., De Paz, J. F., and Corchado, J. M. (2018). A multi-agent system for the classification of gender and age from images. *Computer Vision and Image Understanding*, 172:98–106.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gordo, A., Almazan, J., Revaud, J., and Larlus, D. (2017). End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254.

Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. *Tech. Rep.UCB/CSD-04-1366*.

Grosan, C. and Abraham, A. (2011). Rule-based expert systems. In *Intelligent Systems*, pages 149–185. Springer.

Gruber, I., Hlaváč, M., Železnỳ, M., and Karpov, A. (2017). Facing face recognition with resnet: round one. In *International Conference on Interactive Collaborative Robotics*, pages 67–74. Springer.

Guérin-Dugué, A. and Palagi, P. M. (1994). Texture segmentation using pyramidal gabor functions and self-organising feature maps. *Neural Processing Letters*, 1(1):25–29.

Guo, Z., Zhang, L., and Zhang, D. (2010). A completed modeling of local binary pattern operator for texture classification. *IEEE transactions on image processing*, 19(6):1657–1663.

Habibzadeh, M., Jannesari, M., Rezaei, Z., Baharvand, H., and Totonchi, M. (2018). Automatic white blood cell classification using pre-trained deep learning models: Resnet and inception. In *Tenth International Conference on Machine Vision (ICMV 2017)*, volume 10696, page 1069612. International Society for Optics and Photonics.

Hafizah, W. M., Supriyanto, E., and Yunus, J. (2012). Feature extraction of kidney ultrasound images based on intensity histogram and gray level co-occurrence matrix. In *2012 Sixth Asia Modelling Symposium*, pages 115–120. IEEE.

Hamada, T. (2019). Determinants of decision-makers' attitudes toward industry 4.0 adaptation. *Social Sciences*, 8(5):140.

Hamreras, S., Benítez-Rochel, R., Boucheham, B., Molina-Cabello, M. A., and López-Rubio, E. (2019). Content based image retrieval by convolutional neural networks. In *International Work-Conference on the Interplay Between Natural and Artificial Computation*, pages 277–286. Springer.

Hamreras, S. and Boucheham, B. (2018). Adaptive content based image retrieval based on rice algorithm selection model. In *2018 International Symposium on Programming and Systems (ISPS)*, pages 1–6. IEEE.

Hamreras, S., Boucheham, B., Molina-Cabello, M. A., Benítez-Rochel, R., and López-Rubio, E. (2020). Content based image retrieval by ensembles of deep learning object classifiers. *Integrated Computer-Aided Engineering*, 27(03):317–331.

Han, S., Kang, J., Mao, H., Hu, Y., Li, X., Li, Y., Xie, D., Luo, H., Yao, S., Wang, Y., et al. (2017). Ese: Efficient speech recognition engine with sparse lstm on fpga. In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 75–84.

Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804.

Hayes-Roth, F. (1985). Rule-based systems. *Communications of the ACM*, 28(9):921–932.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

Hemdan, E. E.-D., Shouman, M. A., and Karar, M. E. (2020). Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. *arXiv preprint arXiv:2003.11055*.

Ho, Y.-C. and Pepyne, D. L. (2002). Simple explanation of the no-free-lunch theorem and its implications. *Journal of optimization theory and applications*, 115(3):549–570.

Hoang, T., Do, T.-T., Le Tan, D.-K., and Cheung, N.-M. (2017). Selective deep convolutional features for image retrieval. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1600–1608.

Holland, J. H. (1992a). Complex adaptive systems. *Daedalus*, 121(1):17–30.

Holland, J. H. (1992b). Genetic algorithms. *Scientific american*, 267(1):66–73.

Hong, P., Tian, Q., and Huang, T. S. (2000). Incorporate support vector machines to content-based image retrieval with relevance feedback. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 3, pages 750–753. IEEE.

Höschl IV, C. and Flusser, J. (2016). Robust histogram-based image retrieval. *Pattern Recognition Letters*, 69:72–81.

Hossain, S. I., Akhand, M., Shuvo, M., Siddique, N., and Adeli, H. (2019). Optimization of university course scheduling problem using particle swarm optimization with selective search. *Expert Systems with Applications*, 127:9–24.

Hosseinabadi, A. A. R., Vahidi, J., Saemi, B., Sangaiah, A. K., and Elhoseny, M. (2019). Extended genetic algorithm for solving open-shop scheduling problem. *Soft computing*, 23(13):5099–5116.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hsu, H.-H., Hsieh, C.-W., and Lu, M.-D. (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications*, 38(7):8144–8150.

Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187.

Huang, H.-K., Chiu, C.-F., Kuo, C.-H., Wu, Y.-C., Chu, N. N., and Chang, P.-C. (2016). Mixture of deep cnn-based ensemble model for image retrieval. In *2016 IEEE 5th Global Conference on Consumer Electronics*, pages 1–2. IEEE.

Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., and Zabih, R. (1999). Spatial color indexing and applications. *International Journal of Computer Vision*, 35(3):245–268.

Huang, M.-W., Chen, C.-W., Lin, W.-C., Ke, S.-W., and Tsai, C.-F. (2017). Svm and svm ensembles in breast cancer prediction. *PloS one*, 12(1).

Huberman, B. A. and Clearwater, S. H. (1995). A multi-agent system for controlling building environments. In *ICMAS*, pages 171–176.

Huberman, B. A., Lukose, R. M., and Hogg, T. (1997). An economics approach to hard computational problems. *Science*, 275(5296):51–54.

Humeau-Heurtier, A. (2019). Texture feature extraction methods: A survey. *IEEE Access*, 7:8975–9000.

Husain, S. S. and Bober, M. (2016). Improving large-scale image retrieval through robust aggregation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1783–1796.

Jacovi, A., Shalom, O. S., and Goldberg, Y. (2018). Understanding convolutional neural networks for text classification. *arXiv preprint arXiv:1809.08037*.

Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.

Jayaprakash, A. and Keziselvavijila, C. (2019). Feature selection using ant colony optimization (aco) and road sign detection and recognition (rsdr) system. *Cognitive Systems Research*, 58:123–133.

Jegou, H., Douze, M., and Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. In *European conference on computer vision*, pages 304–317. Springer.

Jindal, A., Dua, A., Kaur, K., Singh, M., Kumar, N., and Mishra, S. (2016). Decision tree and svm-based data analytics for theft detection in smart grid. *IEEE Transactions on Industrial Informatics*, 12(3):1005–1016.

Jing-Ming Guo, Heri Prasetyo (2015). Content-based image retrieval using features extracted from halftoning-based block truncation coding. *IEEE Transactions on Image Processing*.

Jing-Ming Guo, Heri Prasetyo, Nai-Jian Wang (2015). Effective image retrieval system using dot-diffused block truncation coding features. *IEEE Transaction on Multimedia*.

Judd, D. B. (1970). Ideal color space. *Color Eng*, 8(2):36–52.

Kabir, M. M., Islam, M. M., and Murase, K. (2010). A new wrapper feature selection approach using neural network. *Neurocomputing*, 73(16-18):3273–3283.

Kan, M., Shan, S., Chang, H., and Chen, X. (2014). Stacked progressive auto-encoders (spae) for face recognition across poses. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1883–1890.

Kang, J., Park, Y.-J., Lee, J., Wang, S.-H., and Eom, D.-S. (2017). Novel leakage detection by ensemble cnn-svm and graph-based localization in water distribution systems. *IEEE Transactions on Industrial Electronics*, 65(5):4279–4289.

Karamti, H., Tmar, M., Visani, M., Urruty, T., and Gargouri, F. (2018). Vector space model adaptation and pseudo relevance feedback for content-based image retrieval. *Multimedia Tools and Applications*, 77(5):5475–5501.

Kavitha, P. and Prabakaran, S. (2019). A novel hybrid segmentation method with particle swarm optimization and fuzzy c-mean based on partitioning the image for detecting lung cancer.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE.

Khalid, S., Khalil, T., and Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*, pages 372–378. IEEE.

Kishida, K. (2005). *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. National Institute of Informatics Tokyo, Japan.

Ko, A. H., Sabourin, R., and Britto Jr, A. S. (2008). From dynamic classifier selection to dynamic ensemble selection. *Pattern recognition*, 41(5):1718–1731.

Kohavi, R., John, G. H., et al. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324.

Kokkinos, I. (2017). Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6129–6138.

Kolekar, S. V., Pai, R. M., and MM, M. P. (2019). Rule based adaptive user interface for adaptive e-learning system. *Education and Information Technologies*, 24(1):613–641.

Kosko, B. and Toms, M. (1993). *Fuzzy thinking: The new science of fuzzy logic*. Hyperion New York.

Kotthoff, L. (2016). Algorithm selection for combinatorial search problems: A survey. In *Data Mining and Constraint Programming*, pages 149–190. Springer.

Kovarik Jr, V. J. (2006). Cognitive research: Knowledge representation and learning. In *Cognitive Radio Technology*, pages 365–399. Elsevier.

Krawczyk, B., Galar, M., Jeleń, Ł., and Herrera, F. (2016). Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. *Applied Soft Computing*, 38:714–726.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kussul, N., Lavreniuk, M., Skakun, S., and Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Laskar, Z. and Kannala, J. (2017). Context aware query image representation for particular object retrieval. In *Scandinavian Conference on Image Analysis*, pages 88–99. Springer.

Laurini, R. and Thompson, D. (1992). *Fundamentals of spatial information systems*, volume 37. Academic press.

Li, H., Li, Y., and Zha, Y. (2017). Image retrieval method based on multi-view generating and ensemble learning. *International Journal of Performability Engineering*, 13(5).

Li, H., Lin, Z., Shen, X., Brandt, J., and Hua, G. (2015). A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5325–5334.

Li, M., Xu, H., Liu, X., and Lu, S. (2018). Emotion recognition from multichannel eeg signals using k-nearest neighbor classification. *Technology and Health Care*, 26(S1):509–519.

Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., and Chen, M. (2014). Medical image classification with convolutional neural network. In *2014 13th international conference on control automation robotics & vision (ICARCV)*, pages 844–848. IEEE.

Li, Q., Li, W., Wang, J., and Cheng, M. (2019). A sql injection detection method based on adaptive deep forest. *IEEE Access*, 7:145385–145394.

Li, R. (2019). Adaptive learning model based on ant colony algorithm. *International Journal of Emerging Technologies in Learning (iJET)*, 14(01):49–57.

Li, S., Lee, M.-C., and Pun, C.-M. (2008). Complex zernike moments features for shape-based image retrieval. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(1):227–237.

Lin, C.-F. and Wang, S.-D. (2002). Fuzzy support vector machines. *IEEE transactions on neural networks*, 13(2):464–471.

Lin, T.-Y. and Maji, S. (2016). Visualizing and understanding deep texture representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2791–2799.

Lindauer, M., Hoos, H., and Hutter, F. (2015). From sequential algorithm selection to parallel portfolio selection. In *International Conference on Learning and Intelligent Optimization*, pages 1–16. Springer.

Lipton, Z. C. (2018). The mythos of model interpretability. *Queue*, 16(3):31–57.

Liu, C., Chai, K. K., Zhang, X., Lau, E. T., and Chen, Y. (2018). Adaptive blockchain-based electric vehicle participation scheme in smart grid platform. *IEEE Access*, 6:25657–25665.

Liu, D., Hua, K. A., Vu, K., and Yu, N. (2008). Fast query point movement techniques for large cbir systems. *IEEE Transactions on Knowledge and Data Engineering*, 21(5):729–743.

Liu, H., Wang, R., Shan, S., and Chen, X. (2016). Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2064–2072.

Liu, M., He, Y., and Ye, B. (2007). Image zernike moments shape feature evaluation based on image reconstruction. *Geo-spatial Information Science*, 10(3):191–195.

Loesdau, M., Chabrier, S., and Gabillon, A. (2014). Hue and saturation in the rgb color space. In *International conference on image and signal processing*, pages 203–212. Springer.

Long, Y., Gong, Y., Xiao, Z., and Liu, Q. (2017). Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(5):2486–2498.

Loreggia, A., Malitsky, Y., Samulowitz, H., and Saraswat, V. (2016). Deep learning for algorithm portfolios. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Ma, L., Li, M., Gao, Y., Chen, T., Ma, X., and Qu, L. (2017). A novel wrapper approach for feature selection in object-based image classification using polygon-based cross-validation. *IEEE Geoscience and Remote Sensing Letters*, 14(3):409–413.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802.

Markonis, D., Schaer, R., de Herrera, A. G. S., and Müller, H. (2017). The parallel distributed image search engine (paradise). *arXiv preprint arXiv:1701.05596*.

Markonis, D., Schaer, R., and Müller, H. (2016). Evaluating multimodal relevance feedback techniques for medical image retrieval. *Information Retrieval Journal*, 19(1-2):100–112.

Markuš, N., Frljak, M., Pandžić, I. S., Ahlberg, J., and Forchheimer, R. (2014). Eye pupil localization with an ensemble of randomized trees. *Pattern recognition*, 47(2):578–587.

M.E. ElAlami (2014). A new matching strategy for content based image retrieval system. *Applied Soft Computing*.

Messelis, T. and De Causmaecker, P. (2014). An automatic algorithm selection approach for the multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, 233(3):511–528.

Mingqiang, Y., Kidiyo, K., and Joseph, R. (2008). A survey of shape feature extraction techniques. *Pattern recognition*, 15(7):43–90.

Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.

Mohamed, O., Mohammed, O., Brahim, A., et al. (2017). Content-based image retrieval using convolutional neural networks. In *First International Conference on Real Time Intelligent Systems*, pages 463–476. Springer.

Mohanan, A. and Raju, S. (2017). A survey on different relevance feedback techniques in content based image retrieval. *Int. Res. J. Eng. Technol*, 4(2):582–585.

Molina-Cabello, M. A., Luque-Baena, R. M., López-Rubio, E., and Thurnhofer-Hemsi, K. (2018). Vehicle type detection by ensembles of convolutional neural networks operating on super resolved images. *Integrated Computer-Aided Engineering*, 25(4):321–333.

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Mosbah, M. and Boucheham, B. (2015). Selection of relevance feedback technique in the context of cbir. *In The International Arab Conference on Information Technology (ACIT'15)*.

Mosbah, M. and Boucheham, B. (2016). Improving the performance of color-based signatures through dynamic selection of adequate ccv-threshold. In *Proceedings of the 4th Spanish Conference on Information Retrieval*, pages 1–4.

Mosbah, M. and Boucheham, B. (2017a). 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.

Mosbah, M. and Boucheham, B. (2017b). Matching measures in the context of cbir: A comparative study in terms of effectiveness and efficiency. In *World Conference on Information Systems and Technologies*, pages 245–258. Springer.

Mosbah, M. and Boucheham, B. (2017c). Mesures de distance dans le contexte de la recherche d'images par le contenu (CBIR). Ph.D thesis. University of Skikda - 20 Août 1955.

Motoda, H. and Liu, H. (2002). Feature selection, extraction and construction. *Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol*, 5(67-72):2.

Murali, S. and Govindan, V. (2013). Shadow detection and removal from a single image using lab color space. *Cybernetics and information technologies*, 13(1):95–103.

Napoletano, P. (2018). Visual descriptors for content-based retrieval of remote-sensing images. *International Journal of Remote Sensing*, 39(5):1343–1376.

Narendra, K. S. and Annaswamy, A. M. (2012). *Stable adaptive systems*. Courier Corporation.

Nasser, I. M. and Abu-Naser, S. S. (2019). Lung cancer detection using artificial neural network. *International Journal of Engineering and Information Systems (IJEAIS)*, 3(3):17–23.

Nguyen, N.-V., Boucher, A., Ogier, J.-M., and Tabbone, S. (2012). Cluster-based relevance feedback for cbir: a combination of query point movement and query expansion. *Journal of ambient intelligence and humanized computing*, 3(4):281–292.

Noh, H., Araujo, A., Sim, J., Weyand, T., and Han, B. (2017). Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465.

Ojala, T., Pietikainen, M., and 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.

Olmos, R., Tabik, S., and Herrera, F. (2018). Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, 275:66–72.

Owais, M., Arsalan, M., Choi, J., and Park, K. R. (2019). Effective diagnosis and treatment through content-based medical image retrieval (cbmir) by using artificial intelligence. *Journal of clinical medicine*, 8(4):462.

Pang, S., Ma, J., Xue, J., Zhu, J., and Ordonez, V. (2018). Deep feature aggregation and image re-ranking with heat diffusion for image retrieval. *IEEE Transactions on Multimedia*, 21(6):1513–1523.

Pardede, J., Sitohang, B., Akbar, S., and Khodra, M. L. (2017). Comparison of similarity measures in hsv quantization for cbir. In *2017 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6. IEEE.

Pass, G., Zabih, R., and Miller, J. (1997). Comparing images using color coherence vectors. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 65–73.

Peng, X., Zou, C., Qiao, Y., and Peng, Q. (2014). Action recognition with stacked fisher vectors. In *European Conference on Computer Vision*, pages 581–595. Springer.

Perez, C. B. and Olague, G. (2008). Learning invariant region descriptor operators with genetic programming and the f-measure. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE.

Perez, L. and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Persoon, E. and Fu, K.-S. (1977). Shape discrimination using fourier descriptors. *IEEE Transactions on systems, man, and cybernetics*, 7(3):170–179.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE.

Phu, V. N., Tran, V. T. N., Chau, V. T. N., Dat, N. D., and Duy, K. L. D. (2017). A decision tree using id3 algorithm for english semantic analysis. *International Journal of Speech Technology*, 20(3):593–613.

Pietikäinen, M. and Zhao, G. (2015). Two decades of local binary patterns: A survey. In *Advances in independent component analysis and learning machines*, pages 175–210. Elsevier.

Powers, D. M. (2020). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Poynton, C. (1997). Frequently asked questions about color. *Retrieved June*, 19:2004.

Quinlan, J. R. et al. (1996). Bagging, boosting, and c4. 5. In *AAAI/IAAI*, volume 1, pages 725–730.

Radenović, F., Tolias, G., and Chum, O. (2016). Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*, pages 3–20. Springer.

Radenović, F., Tolias, G., and Chum, O. (2018). Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668.

Raghavan, V., Bollmann, P., and Jung, G. S. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems (TOIS)*, 7(3):205–229.

Raju, U., Kumar, A. S., Mahesh, B., and Reddy, B. E. (2010). Texture classification with high order local pattern descriptor: local derivative pattern. *Global Journal of Computer Science and Technology*.

Raman, K., Udupa, R., Bhattacharya, P., and Bhole, A. (2010). On improving pseudo-relevance feedback using pseudo-irrelevant documents. In *European Conference on Information Retrieval*, pages 573–576. Springer.

Ramos, A. R., Acosta, C. D., Torres, P. J. R., Mercado, E. I. S., Baez, G. B., Rifón, L. A., and Llanes-Santiago, O. (2019). An approach to multiple fault diagnosis using fuzzy logic. *Journal of Intelligent Manufacturing*, 30(1):429–439.

Rashno, A. and Sadri, S. (2017). Content-based image retrieval with color and texture features in neutrosophic domain. In *2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pages 50–55. IEEE.

Rautiainen, M., Ojala, T., and Kauniskangas, H. (2001). Detecting perceptual color changes from sequential images for scene surveillance. *IEICE transactions on Information and Systems*, 84(12):1676–1683.

Raza, A., Dawood, H., Dawood, H., Shabbir, S., Mehboob, R., and Banjar, A. (2018). Correlated primary visual texton histogram features for content base image retrieval. *IEEE Access*, 6:46595–46616.

Reinhard, E., Adhikhmin, M., Gooch, B., and Shirley, P. (2001). Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41.

Rice, J. R. (1976). The algorithm selection problem. In *Advances in computers*, volume 15, pages 65–118. Elsevier.

Rizzini, M., Fawcett, C., Vallati, M., Gerevini, A. E., and Hoos, H. H. (2017). Static and dynamic portfolio methods for optimal planning: An empirical analysis. *International Journal on Artificial Intelligence Tools*, 26(01):1760006.

Rocchio, J. (1971). Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323.

Rodriguez-Galiano, V., Luque-Espinar, J., Chica-Olmo, M., and Mendes, M. (2018). Feature selection approaches for predictive modelling of groundwater nitrate pollution: An evaluation of filters, embedded and wrapper methods. *Science of the total environment*, 624:661–672.

Rokach, L., Schclar, A., and Itach, E. (2014). Ensemble methods for multi-label classification. *Expert Systems with Applications*, 41(16):7507–7523.

Roussel, O. (2012). Description of ppfolio (2011). *Proc. SAT Challenge*, page 46.

Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., and Beling, P. (2018). Deep learning detecting fraud in credit card transactions. In *2018 Systems and Information Engineering Design Symposium (SIEDS)*, pages 129–134. IEEE.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Sajja, P. S. and Akerkar, R. (2013). Bio-inspired models for semantic web. In *Swarm Intelligence and Bio-Inspired Computation*, pages 273–294. Elsevier.

Saka, M. P., Doğan, E., and Aydogdu, I. (2013). Analysis of swarm intelligence–based algorithms for constrained optimization. In *Swarm Intelligence and Bio-Inspired Computation*, pages 25–48. Elsevier.

Sánchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

Saritha, R. R., Paul, V., and Kumar, P. G. (2019). Content based image retrieval using deep learning process. *Cluster Computing*, 22(2):4187–4200.

Sato, H. and Narita, R. (2013). Efficient maximum range search on remote spatial databases using k-nearest neighbor queries. *Procedia Computer Science*, 22:836–845.

Schalkoff, R. J. (2007). Pattern recognition. *Wiley Encyclopedia of Computer Science and Engineering*.

Schapire, R. E. (2013). Explaining adaboost. In *Empirical inference*, pages 37–52. Springer.

Schütze, H., Manning, C. D., and Raghavan, P. (2009). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Shafarenko, L., Petrou, H., and Kittler, J. (1998). Histogram-based segmentation in a perceptually uniform color space. *IEEE transactions on image processing*, 7(9):1354–1358.

Shang, W., Sohn, K., Almeida, D., and Lee, H. (2016). Understanding and improving convolutional neural networks via concatenated rectified linear units. In *international conference on machine learning*, pages 2217–2225.

Shen, D. and Ip, H. H. (1999). Discriminative wavelet shape descriptors for recognition of 2-d patterns. *Pattern recognition*, 32(2):151–165.

Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., Yang, B., and Liu, D. (2016). Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems*, 96:61–75.

Shriram, K., Priyadarsini, P., and Baskar, A. (2015). An intelligent system of content-based image retrieval for crime investigation. *International Journal of Advanced Intelligence Paradigms*, 7(3/4):264–279.

Silva, V. S., Freitas, A., and Handschuh, S. (2019). On the semantic interpretability of artificial intelligence models. *arXiv preprint arXiv:1907.04105*.

Simon, P. (2013). *Too big to ignore: the business case for big data*, volume 72. John Wiley & Sons.

Simonyan, K., Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2013). Fisher vector faces in the wild. In *BMVC*, volume 2, page 4.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, V., Sarwar, A., and Sharma, V. (2017). Analysis of soil and prediction of crop yield (rice) using machine learning approach. *International Journal of Advanced Research in Computer Science*, 8(5).

Smith-Miles, K. A. (2009). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys (CSUR)*, 41(1):6.

Srinivasan, G. and Shobha, G. (2008). Statistical texture analysis. In *Proceedings of world academy of science, engineering and technology*, volume 36, pages 1264–1269.

Steiger, R. and Scott, D. (2020). Ski tourism in a warmer world: Increased adaptation and regional economic impacts in austria. *Tourism Management*, 77:104032.

Stricker, M. A. and Orengo, M. (1995). Similarity of color images. In *Storage and retrieval for image and video databases III*, volume 2420, pages 381–392. International Society for Optics and Photonics.

Sugianto, N., Tjondronegoro, D., and Tydd, B. (2018). Deep residual learning for analyzing customer satisfaction using video surveillance. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE.

Sultani, W., Chen, C., and Shah, M. (2018). Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6479–6488.

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International journal of computer vision*, 7(1):11–32.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Tan, X. and Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, 19(6):1635–1650.

Tang, D., Wei, F., Qin, B., Liu, T., and Zhou, M. (2014). Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 208–212.

Teague, M. R. (1980). Image analysis via the general theory of moments. *JOSA*, 70(8):920–930.

Teh, C.-H. and Chin, R. T. (1988). On image analysis by the methods of moments. *IEEE Transactions on pattern analysis and machine intelligence*, 10(4):496–513.

Tortorella, R. A. and Graf, S. (2017). Considering learning styles and context-awareness for mobile adaptive learning. *Education and Information Technologies*, 22(1):297–315.

Trefnỳ, J. and Matas, J. (2010). Extended set of local binary patterns for rapid object detection. In *Computer vision winter workshop*, pages 1–7.

Truong, H. M. (2016). Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities. *Computers in human behavior*, 55:1185–1193.

Tsai, C.-F. (2012). Bag-of-words representation in image annotation: A review. *ISRN Artificial Intelligence*, 2012.

Tsamardinos, I. and Aliferis, C. F. (2003). Towards principled feature selection: relevancy, filters and wrappers. In *AISTATS*.

Tulsian, V., Kanade, A., Kumar, R., Lal, A., and Nori, A. V. (2014). Mux: algorithm selection for software model checkers. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 132–141.

Turner, M. R. (1986). Texture discrimination by gabor functions. *Biological cybernetics*, 55(2-3):71–82.

Tzelepi, M. and Tefas, A. (2016). Relevance feedback in deep convolutional neural networks for content based image retrieval. In *Proceedings of the 9th Hellenic Conference on Artificial Intelligence*, pages 1–7.

Uthayakumar, J., Metawa, N., Shankar, K., and Lakshmanaprabu, S. (2020). Financial crisis prediction model using ant colony optimization. *International Journal of Information Management*, 50:538–556.

Vadivel, A., Majumdar, A., and Sural, S. (2003). Performance comparison of distance metrics in content-based image retrieval applications. In *International Conference on Information Technology (CIT), Bhubaneswar, India*, pages 159–164.

Van Gool, L., Dewaele, P., and Oosterlinck, A. (1985). Texture analysis anno 1983. *Computer vision, graphics, and image processing*, 29(3):336–357.

Vasan, D., Alazab, M., Wassan, S., Safaei, B., and Zheng, Q. (2020). Image-based malware classification using ensemble of cnn architectures (imcec). *Computers & Security*.

Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., and Li, J. (2014a). Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 157–166.

Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., and Li, J. (2014b). Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 157–166. ACM.

Wang, J., Kumar, S., and Chang, S.-F. (2010). Semi-supervised hashing for scalable image retrieval. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3424–3431. IEEE.

Wang, X.-Y., Liang, L.-L., Li, W.-Y., Li, D.-M., and Yang, H.-Y. (2016). A new svm-based relevance feedback image retrieval using probabilistic feature and weighted kernel function. *Journal of Visual Communication and Image Representation*, 38:256–275.

Wei, L.-Y., Lefebvre, S., Kwatra, V., and Turk, G. (2009). State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*, pages 93–117. Eurographics Association.

Wei, X.-S., Luo, J.-H., Wu, J., and Zhou, Z.-H. (2017). Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Transactions on Image Processing*, 26(6):2868–2881.

Wen, G., Chen, H., Cai, D., and He, X. (2018). Improving face recognition with domain adaptation. *Neurocomputing*, 287:45–51.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

Wyman, C., Sloan, P.-P., and Shirley, P. (2013). Simple analytic approximations to the cie xyz color matching functions. *Journal of Computer Graphics Techniques*, 2(2):1–11.

Xia, R., Pan, Y., Lai, H., Liu, C., and Yan, S. (2014). Supervised hashing for image retrieval via image representation learning. In *Twenty-eighth AAAI conference on artificial intelligence*.

Xiang, X., Peng, Y., and Zhang, L. (2009). A method of optical flow computation based on luv color space. In *2009 International Conference on Test and Measurement*, volume 2, pages 378–381. IEEE.

Xie, L., Shen, J., Han, J., Zhu, L., and Shao, L. (2017). Dynamic multi-view hashing for online image retrieval. IJCAI.

Xu, D., Ricci, E., Yan, Y., Song, J., and Sebe, N. (2015). Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*.

Xu, H., Wang, J.-y., and Mao, L. (2017). Relevance feedback for content-based image retrieval using deep learning. In *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pages 629–633. IEEE.

Xu, J., Shi, C., Qi, C., Wang, C., and Xiao, B. (2018a). Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Xu, J., Wang, C., Qi, C., Shi, C., and Xiao, B. (2018b). Unsupervised semantic-based aggregation of deep convolutional features. *IEEE Transactions on Image Processing*, 28(2):601–611.

Xu, J., Xu, C., Zou, B., Tang, Y. Y., Peng, J., and You, X. (2018c). New incremental learning algorithm with support vector machines. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11):2230–2241.

Xu, L., Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2008). Satzilla: portfolio-based algorithm selection for sat. *Journal of artificial intelligence research*, 32:565–606.

Yang, B., Shang, X., and Pang, S. (2017). Isometric hashing for image retrieval. *Signal Processing: Image Communication*, 59:117–130.

Yang, F., Matei, B., and Davis, L. S. (2015). Re-ranking by multi-feature fusion with diffusion for image retrieval. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 572–579. IEEE.

Yang, J., Liang, J., Shen, H., Wang, K., Rosin, P. L., and Yang, M.-H. (2018). Dynamic match kernel with deep convolutional features for image retrieval. *IEEE Transactions on Image Processing*, 27(11):5288–5302.

Yang, J., Liu, C., and Zhang, L. (2010). Color space normalization: Enhancing the discriminating power of color spaces for face recognition. *Pattern Recognition*, 43(4):1454–1466.

Yang, X.-S. and Karamanoglu, M. (2013). Swarm intelligence and bio-inspired computation: an overview. In *Swarm intelligence and bio-inspired computation*, pages 3–23. Elsevier.

Yang, Y., Wang, J., Wang, G., and Chen, Y.-W. (2019). Research and development project risk assessment using a belief rule-based system with random subspaces. *Knowledge-Based Systems*, 178:51–60.

Yao, K., Cohn, T., Vylomova, K., Duh, K., and Dyer, C. (2015). Depth-gated lstm. *arXiv preprint arXiv:1508.03790*.

Yesubai Rubavathi Charles, Ravi Ramraj (2016). A novel local mesh color texture pattern for image retrieval system. *International Journal of Electronics and Communications*.

Yin, C., Zhu, Y., Fei, J., and He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, 5:21954–21961.

Younes, L. (1999). Optimal matching between shapes via elastic deformations. *Image and Vision Computing*, 17(5-6):381–389.

Yu, S., Wickstrøm, K., Jenssen, R., and Principe, J. C. (2020). Understanding convolutional neural networks with information theory: An initial exploration. *IEEE Transactions on Neural Networks and Learning Systems*.

Zadeh, L. A. (1988). Fuzzy logic. *Computer*, 21(4):83–93.

Zareapoor, M., Shamsolmoali, P., et al. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia computer science*, 48(2015):679–685.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Zhang, B., Wu, Y., Lu, J., and Du, K.-L. (2011). Evolutionary computation and its applications in neural and fuzzy systems. *Applied Computational Intelligence and Soft Computing*, 2011.

Zhao, G. and Pietikainen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):915–928.

Zhi, H. and Liu, S. (2019). Face recognition based on genetic algorithm. *Journal of Visual Communication and Image Representation*, 58:495–502.

Zhou, J. T., Du, J., Zhu, H., Peng, X., Liu, Y., and Goh, R. S. M. (2019). Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, 14(10):2537–2550.

Zhou, Z.-H. (2009). Ensemble learning. *Encyclopedia of biometrics*, 1:270–273.

Zhu, L., Shen, J., Xie, L., and Cheng, Z. (2016). Unsupervised topic hypergraph hashing for efficient mobile image retrieval. *IEEE transactions on cybernetics*, 47(11):3941–3954.

Zhu, Y., Jiang, J., Han, W., Ding, Y., and Tian, Q. (2017). Interpretation of users' feedback via swarmed particles for content-based image retrieval. *Information Sciences*, 375:246–257.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.