الجمهورية الجزائرية الديموقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة 20 أوت 1955- سكيكدة

كلية العلوم

قسم الإعلام الآلي

Master's degree dissertation

Major: Computer Science

Minor: Artificial Intelligence (AI)

Title:

Image Classification as an Image Mining Task: A Machine Learning Approach

Presented by:

KIFADJI Taissir El Amel

Supervised by:

Pr. BOUCHEHAM Bachir

Session June, 2025

Acknowledgments

I would like to express my sincere thanks to the administration of my university (University of Skikda, 20 August 1955), the Faculty of Science, as well as the heads of the Computer Science Department for providing me with a favorable environment and the necessary conditions throughout my academic journey.

I also wish to express my deep gratitude to my supervisor and thesis advisor, **Prof. Bachir Boucheham**, for his expertise, availability, and invaluable guidance. He gave me the opportunity to explore the field of Data Mining, and more specifically, Image Mining. This thesis would not have been possible without his unwavering support.

My warmest thoughts go to all the people dear to my heart for their support, love, advice, help, and for always being by my side.

Finally, I would like to thank all those who supported me, whether directly or indirectly, throughout the completion of this work.

Abstract

In today's digital world, the volume of data textual, visual, and audio has increased exponentially, significantly transforming the way information is processed and analyzed across domains. This massive growth has driven researchers to explore advanced methods for extracting useful knowledge, giving rise to the field of Data Mining. Initially focused on structured data from relational databases, data mining has evolved to handle complex data types such as images, videos, and data from social networks, ushering in the Big Data era. Big Data is characterized by five key properties: volume, variety, velocity, veracity, and value each requiring tailored techniques for effective processing.

This thesis focuses on image classification, a core task within the specialized field of Image Mining, which itself extends traditional data mining techniques to visual content. Image mining involves extracting meaningful patterns from large image collections by analyzing their visual characteristics and semantic content. These patterns can then be used for various tasks, such as classification, similarity detection, and anomaly identification.

To address the image classification problem, this research explores two primary approaches:

- Machine learning, using classifiers such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Naive Bayes, Decision Trees, and Random Forests.
- Deep learning, particularly Convolutional Neural Networks (CNNs), which automatically learn hierarchical representations from image data.

A comparative study was conducted using two datasets with distinct characteristics:

- Corel-1K, a general-purpose dataset of 1,000 images grouped into 10 semantic categories,
- KimiaPath960, a medical dataset composed of digital pathology images.

The experiments tested the performance of the classifiers in terms of accuracy and execution time using five feature extraction techniques: Local Binary Patterns (LBP), Haralick descriptors, HSV histograms, Color Moments, and Fourier Descriptors.

The key findings are:

1. The choice of classifier, feature type, color space, and dataset structure all significantly influence classification performance.

2. CNNs were more effective on semantically rich images (Corel-1K) than on semantically poor ones (KimiaPath960).
3. Certain images are inherently easier to classify due to the richness or simplicity of their visual and semantic content.

This research contributes to a deeper understanding of how intelligent systems can extract and classify visual data effectively using both classical and deep learning-based techniques.

الملخص

في عالمنا الرقمي اليوم، شهد حجم البيانات سواء النصية أو المرئية أو السمعية نموًا هائلًا، مما غيّر بشكل كبير طريقة معالجة وتحليل المعلومات في مختلف المجالات. وقد دفع هذا النمو المتسارع الباحثين إلى استكشاف أساليب متقدمة لاستخلاص المعرفة المفيدة، مما أدى إلى ظهور مجال **تنقيب البيانات** .ركز هذا المجال في بداياته على البيانات المهيكلة المستخرجة من قواعد البيانات العلائقية، لكنه تطور لاحقًا ليشمل أنواعًا أكثر تعقيدًا من البيانات، مثل الصور ومقاطع الفيديو وبيانات شبكات التواصل، مما مهد الطريق لعصر البيانات الضخمة(Big Data) ، التي تتميز بخمس خصائص رئيسية: الحجم، التنوع، السرعة، الموثوقية، والقيمة. وهذه الخصائص تتطلب تقنيات تحليل ملائمة وفعالة.

يركز هذا البحث على تصنيف الصور، وهو مهمة أساسية ضمن فرع متخصص من تنقيب البيانات يُعرف بـ **تنقيب الصور** .**(Image Mining)** يهدف هذا المجال إلى استخراج أنماط ذات دلالة من مجموعات كبيرة من الصور، من خلال تحليل تمثيلاتها البصرية ومحتواها الدلالي. وتُستخدم هذه الأنماط في مهام متعددة، مثل التصنيف، والبحث عن التشابه، واكتشاف الحالات الشاذة.

لمعالجة مهمة تصنيف الصور، يستكشف هذا العمل مقاربتين رئيسيتين:

- **التعلم الآلي (Machine Learning)** باستخدام مصنفات مثل الجار الأقرب(KNN) ، وآلات الدعم الناقل(SVM) ، ونايف بايز، وأشجار القرار، والغابة العشوائية.
- **التعلم العميق(Deep Learning)** ، خاصة **الشبكات العصبية الالتفافية(CNNs)** ، المعروفة بقدرتها على تعلم تمثيلات هرمية معقدة من الصور تلقائيًا.

تم إجراء دراسة مقارنة باستخدام مجموعتي بيانات مختلفتين:

- **Corel-1K**: مجموعة بيانات عامة تحتوي على 1000 صورة موزعة على 10 فئات دلالية،
- **KimiaPath960**: مجموعة بيانات طبية تحتوي على صور رقمية لشرائح نسيجية مرضية.

وقد شملت التجارب تقييم أداء كل خوارزمية من حيث **دقة التصنيف** و**زمن التنفيذ**، وذلك باستخدام تقنيات مختلفة لاستخلاص الميزات من الصور، مثل: الأنماط الثنائية المحلية(LBP) ، وواصفاتHaralick ، وهيستوغرامHSV ، ولحظات الألوان(Color Moments) ، والمواصفات الفوريه (Fourier Descriptors).

وقد أظهرت النتائج الرئيسية ما يلي:

1. تؤثر جميع العوامل (نوع المصنف، نوع السمات، فضاء الألوان، وهيكلية البيانات) بشكل كبير على نتائج التصنيف.
2. أظهرت الشبكات العصبية الالتفافية أداءً أفضل مع الصور الغنية دلاليًا (Corel-1K) مقارنةً بالصور الفقيرة دلاليًا.(KimiaPath960)
3. هناك صور أسهل في التصنيف من غيرها، وذلك حسب درجة تعقيد أو بساطة محتواها الدلالي.

يساهم هذا البحث في فهم أعمق لكيفية استخدام الأنظمة الذكية لاستخلاص وتصنيف المعلومات البصرية بشكل فعال، سواء باستخدام خوارزميات تقليدية أو تقنيات التعلم العميق.

Key words: Data Mining, Image Mining, Image Classification, Image Characterization, Image Comparaison, Machine Learning.

Table of Contents

Table of Figures

Chapter 1: Data Mining: General Introduction and Literature Overview

1. Introduction

In exploring the fundamentals of our field of study, this first chapter offers a general introduction to data mining, its origins and evolution. The paradigm of data mining initially emerged from the need to extract useful patterns and knowledge from structured data stored in databases. As databases evolved into data warehouses designed to organize and integrate large volumes of data new opportunities for deeper data exploration emerged, paving the way for data mining as a key process in data analysis.

Over time, data mining has extended beyond structured data to include unstructured data, contributing significantly to the emergence of big data. In this context, data mining plays a crucial role within the broader process of Knowledge Discovery in Databases (KDD), serving as the central step that extracts actionable insights from vast and complex datasets.

Finally, this chapter emphasizes the two principal tasks of data mining: descriptive tasks and predictive tasks, both are essential in supporting informed decision-making in various domains.

2. Data Mining Period

Before the emergence of the data mining paradigm, data was managed using databases and analyzed using data warehouses.

2.1 Data Bases

Since the 1960s, information processing technologies have evolved significantly, transitioning from simple file management systems to sophisticated Database Management Systems (DBMS). A database is an organized structure that stores information in tables consisting of rows and columns. This tabular format enables efficient querying, manipulation, and management of data using SQL (Structured Query Language). DBMSs are responsible for storing, organizing data, easy access and management.

In this context, Online Transaction Processing (OLTP) systems play a crucial role by allowing processing of large volumes of real-time transactions or online banking services. However, as companies expanded and established branches across different regions, it became increasingly difficult to make decisions by relying on scattered operational databases, OLTP systems are not designed to handle complex queries. As data grows in volume and originates from diverse sources, processing each source separately becomes

costly. This challenge prompted researchers to develop centralized systems capable of analyzing heterogeneous data, named data warehouses.

2.2 Data Warehouses

A Data Warehouse is a centralized database that integrates multiple operational databases. **[1]** While daily operations use OLTP systems for managing transactions, data warehouses are designed for analytical purposes and are typically updated at regular intervals rather than in real time **[2]**. Data Warehousing technology is a set of operations specific to data warehouses, used to analyze data stored within them. In this context, one of the first tools is the data cube, a multidimensional structure that enables data exploration. It supports powerful operations such as slicing (selecting a single dimension), dicing (selecting a subcube), drill-down (viewing data at a more detailed level), and roll-up (aggregating data to a higher level) **[3].** In addition to data cubes, the technology includes key steps in the Knowledge Discovery in Databases (KDD) process, such as data cleaning, data integration, and data selection. Also using Online Analytical Processing (OLAP) techniques for business intelligence.

3. Definition of Data Mining

Data Mining is an iterative and interactive process aimed at discovering new, useful, understandable, and time-valid knowledge from large datasets [Fayyad et al., 1995]. It involves applying scientific methods to explore, analyze, and extract patterns, recurring behaviors, governing rules, unknown trends, and meaningful structures from complex and heterogeneous data [TUFFERY, 2014]. This knowledge aids decision-making processes.

4. Objectives of data mining

- **Understanding Current Data Behavior:**
  The first objective of Data Mining is to better understand the current state of data. By discovering existing patterns.
- **Predicting Future Data Behavior:**
  The second goal is to forecast future trends based on historical and current data. By identifying trends and patterns, Data Mining helps predict future outcomes, which may enable businesses to take proactive actions.

5. Data Mining typology

The data mining paradigm emerged in structured and multimedia data.

5.1 Structured Data (in Data Bases)

Data Mining of structured data refers to the process of discovering patterns, relationships, and trends in organized datasets, typically stored in relational databases. This form of Data Mining was the first to emerge, making it the most widely used approach. It involves working with data that is organized into tables, allowing for easier management.

The development of structured data mining was significantly motivated by practical cases, notably the famous **"Market Basket Analysis"** or **"Cash Register Receipts"** case. This particular case focused on extracting useful insights to optimize the management of American supermarkets **[4]**. By mining transaction data, retailers were able to uncover patterns in product purchases, which led to better inventory control, targeted promotions, and improved customer service.

Thus, structured data mining focuses on extracting valuable knowledge from various data sources such as transactional records, customer databases, and sales data. This knowledge is then used to support business decision-making, helping organizations identify hidden patterns and trends.

5.2 Unstructured Data (Multimedia-Complex Data)

Over the years, Data Mining has significantly evolved to address the growing challenges posed by the diversity and complexity of available data. In its early stages, the field primarily focused on data organized into tables in relational databases. However, with the explosion of data volume and variety generated worldwide, researchers and data scientists have expanded the scope of Data Mining to include unstructured data, these are text, images, audio, video, time series, web pages. These data types, while rich in information, present unique challenges due to their complex nature.

One of the main challenges associated with unstructured data lies in the process of extracting relevant features. Unlike structured data, that use conventional techniques, unstructured data requires specific approaches to extract meaningful features. In this context, Content-Based Data Mining plays a central role, focusing on extracting attributes directly from the content of the data rather than relying on metadata.

Thus, Data Mining, especially when applied to unstructured data, provides opportunities to discover patterns, trends, and anomalies that would be difficult to perceive. The application of these techniques may be used in various fields, such as image analysis for visual

recognition, text mining for information retrieval, and the analysis of biological signals for medical applications.

5.3 Big Data

Data mining initially focused on structured data. As technology advanced, the scope of data mining expanded to include multimedia data such as images, audio, video, and text, within social networks such as Facebook, Twitter, Instagram and LinkedIn. This evolution contributed to the emergence of Big Data, which addresses the challenges of managing and analyzing vast, diverse, and rapidly evolving datasets. (in V1, V2, V3, V4, V5) that will be explained further below.

5.4 Characteristics of Big Data :(5 V's)

There are five fundamental characteristics of Big Data, commonly known as the 5 V's. However, more may be added over time as the field continues to evolve.

V1: Volume: One of the primary features of Big Data is the sheer volume of data being generated. Data is collected from a variety of sources, such as business transactions, sensors, social media, and more, leading to massive datasets. As the amount of data continues to increase, organizations must develop strategies for storing, managing, and processing this vast quantity.

V2: Velocity: Velocity refers to the speed at which data is generated and the rate at which it must be processed. With Big Data, data is often produced in real-time, necessitating the use of advanced technologies capable of processing data quickly. The faster the data is processed, the more value can be extracted in a timely manner, that's essential for making informed decisions for financial markets, social media monitoring, or real-time analytics.

V3: Variety: Big Data encompasses a wide variety of data types, both structured and unstructured. Structured data such as databases, while unstructured data can include text, images, audio, video, and social media posts. The complexity of managing and integrating these diverse data types adds to the challenge, requiring tools and techniques that can handle different formats, structures, and sources.

V4: Veracity: Veracity refers to the quality, accuracy, and reliability of the data. With such a massive volume and variety of data, ensuring that it is trustworthy becomes critical. Data from different sources can vary in terms of completeness,

consistency, and reliability, which can affect the overall quality of the insights derived. Addressing issues like missing data, data noise, and inaccuracies is essential for maintaining the integrity of the data and ensuring valid outcomes.

V5: Value: the purpose of Big Data is to extract meaningful, actionable insights that provide value to organizations. The value lies not just in collecting large amounts of data but in the ability to process and analyze it effectively to uncover patterns, trends, and correlations that inform business decisions, drive innovation.

6. Knowledge Discovery in Data (KDD) Process

This section presents the KDD process step by step and highlights the central role of the data mining phase within the overall knowledge discovery**.**

6.1 Definition of KDD

Knowledge Discovery in Databases (KDD) is the process of extracting valuable insights from large datasets using techniques from computer science, mathematics, statistics, and artificial intelligence. The goal is to transform raw data into actionable knowledge that can support decision-making. The KDD process involves several stages: data selection, cleaning, transformation, mining, and the interpretation of discovered patterns. **[5]**

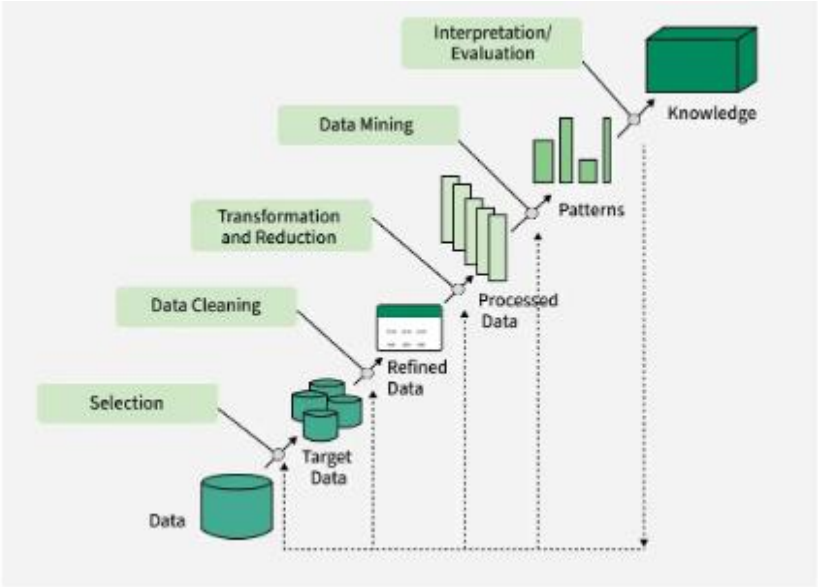

Figure1 Diagram of the steps of the KDD process **[6]**

6.2 The Steps of KDD Process

The steps of the KDD process include eight essential phases that guide the transformation of raw data into useful knowledge.

6.2.1 Data selection

Data selection is a fundamental step in any analytical process, as it directly influences the quality of the outcomes. It involves identifying and retaining only the most relevant data based on the study's objectives, while filtering out unnecessary, incomplete, or irrelevant information. This approach not only enhances processing efficiency but also ensures the reliability and validity of subsequent analyses.

6.2.2 Data Cleaning

Data cleaning is a crucial preprocessing step that involves removing noise and irrelevant data from a large dataset. The quality of the selected data directly impacts the outcome of the analysis. This process includes eliminating duplicate records, handling missing values by entering correct data, discarding unnecessary data, standardizing data formats, and ensuring timely updates. Proper data cleaning enhances the accuracy of the results.**[7]**

6.2.3 Data Integration

Data integration involves merging data from multiple databases or text files into a unified database. In data mining, relevant information often originates from various sources rather than a single database. This process is carried out using unique attributes such as names, product types, or customer numbers to ensure accurate entity identification. However, data integration must be handled carefully, as errors can lead to misleading results. For instance, if products from different categories are mistakenly combined based on product type, false correlations may arise, affecting the accuracy of the result.**[8]**

6.2.4 Data Transformation

Data transformation involves applying various methods to convert selected data into an appropriate format for analysis. This step may include dimensionality reduction or transformation techniques to reduce the number of variables while retaining only the most useful features.

6.2.5 Mining Process

The mining process is a crucial stage where selected data mining methods are applied to extract valuable and hidden knowledge from data. This step involves identifying meaningful patterns, relationships, and insights that can support decision-making and predictions with enhancing the overall understanding of the dataset.

6.2.6 Evaluate Patterns

Pattern evaluation is a crucial step in the data mining process, aimed at analyzing the relevance, validity of the relationships discovered by the model. It helps determine whether the extracted patterns align with the initial. Various evaluation metrics such as accuracy, F-measure are used to assess the model's ability to classify or predict outcomes based on the training data. This evaluation also verifies the statistical robustness of the patterns, identifies possible redundancies or biases, and assesses their practical value within the study's context. If the results are unsatisfactory, adjustments can be made by refining parameters, exploring alternative methods.

6.2.7 Knowledge Presentation

Knowledge presentation is a crucial step in the data mining process, ensuring that the extracted insights are easily interpretable and accessible to users. Instead of treating the application as a "black box," the results should be presented in a clear and understandable format, often using visualization techniques to aid interpretation. Effective knowledge representation helps users comprehend the discovered patterns and supports informed decision-making. The overall data mining process, from data selection and preprocessing to transformation, mining, evaluation, and interpretation, ultimately leads to the extraction of valuable knowledge that can drive meaningful actions.**[8]**

6.2.8 Interpretation and Deployment

Interpretation of results involves analyzing and refining the knowledge extracted from models to translate it into actionable insights. This step includes techniques such as identifying important variables, visualizing results, and applying methods like SHAP (SHapley Additive exPlanations) to explain the decisions made by complex models **[9].** Once the results are interpreted, they can be deployed into real-world applications or decision-making processes, such as integration into automated systems or decision support tools. Continuous monitoring of the model's performance under real-world conditions is essential to adapt solutions to new data or evolving contexts.

6.3 The role of Data Mining in KDD

Data mining is a crucial phase within the Knowledge Discovery in Databases (KDD) process, where various techniques are applied to extract valuable patterns and insights from data. While stages like data selection, preprocessing, and interpretation are important for preparing and refining data, it is during the mining process that these stages come together to generate knowledge. Data mining focuses on uncovering hidden, non-obvious patterns in structured, semi-structured, or unstructured data, which can be used for predictions, classifications, and discovering correlations. For example, in medical datasets, data mining techniques can identify patterns of disease progression or predict

patient outcomes, while in retail, mining customer transaction data can reveal purchasing trends or product recommendations. Techniques such as classification, clustering, association rule mining, and regression are applied during the mining process to reveal deep insights. These techniques transform raw data into meaningful insights, enabling more accurate predictions and informed decision-making.

7. Data Mining Tasks [10]

Data mining tasks are divided into two primary categories: descriptive tasks and predictive tasks.

7.1 Descriptive Tasks

Descriptive tasks are primarily focused on summarizing the key characteristics of a dataset. These tasks allow users to identify patterns and relationships that are not immediately obvious due to the vast amount of data. Descriptive tasks do not involve prediction but instead focus on understanding and revealing the structure and distribution of the data.

Key objectives of descriptive tasks include:

- Identifying patterns: This includes detecting trends, outliers, or associations between variables within the dataset.
- Summarizing data: Through methods such as clustering, or dimensionality reduction
- Data visualization: Presenting the data through charts, graphs, and visual summaries.

Examples of descriptive tasks include:

- Data segmentation: Grouping data points into clusters based on similar attributes or behaviors, such as customer segmentation in marketing.
- Association rule mining: Identifying frequent patterns or associations between different data attributes, such as discovering that people who buy milk are likely to buy bread (market basket analysis).
- Data summarization and reduction: Reducing the size of large datasets while retaining important information, such as using Principal Component Analysis (PCA) or other dimensionality reduction techniques to simplify high-dimensional data.

7.2 Predictive Tasks

Predictive tasks focus on using the data to predict future outcomes or classify data into categories based on patterns observed in the past. These tasks go beyond just

understanding the data and aim to provide actionable insights that can be used for decision-making. Predictive tasks often rely on machine learning algorithms, statistical models.

Key objectives of predictive tasks include:

1. Building models: Using historical data to create models that can predict future events or behaviors. These models are then validated and tested for accuracy using evaluation metrics.
2. Forecasting trends: Predicting future values or trends, such as sales forecasts, stock market predictions, or weather forecasts.

Examples of predictive tasks include:

- Classification: This involves assigning data points to one of several predefined classes or categories. For example, classifying emails as either "spam" or "not spam" or categorizing customer behavior into "high risk" or "low risk."
- Regression: Predicting continuous values based on historical data. For instance, predicting the future price of a stock, the temperature on a given day, or a person's income based on various factors.
- Anomaly detection: Detecting outliers or anomalies in the data that deviate significantly from the normal patterns. This can be used for fraud detection, network security, and quality control.

8. Conclusion

In summary, this chapter has led the groundwork for understanding data mining by tracing its evolution from traditional databases to data warehouses, and ultimately to the era of big data. We highlighted how the growing complexity and volume of data have necessitated more advanced methods for uncovering patterns and extracting knowledge. Data mining has emerged as a pivotal step in transforming raw data into valuable insights.

Furthermore, the chapter clarified the distinction between the two core tasks of data mining descriptive and predictive analysis both of which play a fundamental role in enhancing decision-making processes across a wide range of application domains. This general overview sets the stage for a deeper exploration of data mining techniques, algorithms, and their applications, which will be discussed in the following chapters.

.

Chapter 2: Image Mining as a subfield of Multimedia Mining and Data Mining

1.Introduction

In the ever-expanding field of multimedia mining and data mining, image mining has gained significant importance due to the vast amounts of image data generated daily across various industries. As a subfield, image mining specifically focuses on extracting valuable information and patterns from images, which can be considered unstructured data. This chapter provides an exploration of the fundamental concepts of image mining, outlining the critical steps required to transform raw image data into useful insights.

The chapter begins by addressing the basic concepts of an image, offering a detailed understanding of what an image is, how it is represented (including binary, grayscale, and color formats), and how these representations influence image analysis techniques. This foundational knowledge is crucial for understanding the underlying principles of image mining.

Next, we explore the essential image features, such as texture, color, shape, and patterns, which are key to identifying and analyzing the content of images. These features serve as the building blocks for various image mining tasks.

2. Basic Concepts of Images

Images are essential data structures in various fields. Understanding the fundamental concepts of images is crucial for effectively working with them in these domains. This section explores the basic concepts involved in defining, representing, and extracting features from images.

2.1 Definition of an Image

An image is a set of visual data organized to represent either a reality or an abstraction. It can be produced through various technical or creative processes and can take different forms, ranging from simple graphics to more complex representations. An image can be static showing a single, or animated consisting of a sequence of images shown over time to convey motion or change such as in video frames or time-lapse medical scans and its function varies depending on its context, creation, and use. It allows for the transmission of visual information. **[11]**

2.2 Image Representation

Image representation refers to the method in which visual data is stored, organized, and processed in digital form. In digital image processing, images are typically represented as matrices of pixel values. Each pixel, which is the smallest unit of an image, contains information such as color, brightness, and sometimes transparency.

Transparency is managed through an additional component called the alpha channel, which complements the standard red, green, and blue (RGB) channels to form the RGBA model. The alpha channel defines the level of opacity for each pixel: a value of 0 indicates full transparency, while a maximum value (e.g., 255 in 8-bit representation) indicates full opacity. This feature allows for the blending of images and is widely used in digital image compositing, graphic design, and visual effects. **[12]** For example, in medical imaging, the alpha channel can be used to overlay diagnostic information (such as tumor contours or segmented organs) on top of original scans, enabling radiologists and clinicians to visualize both raw and processed data without losing context. The organization of these pixels in rows and columns allows for the creation of a digital image.

2.2.1. Binary Representation:

In a binary image, each pixel has only two possible values: **0** (black) or **1** (white). This type of representation is often used in simple images, such as document images or text images, where only two colors are needed to distinguish objects from the background.

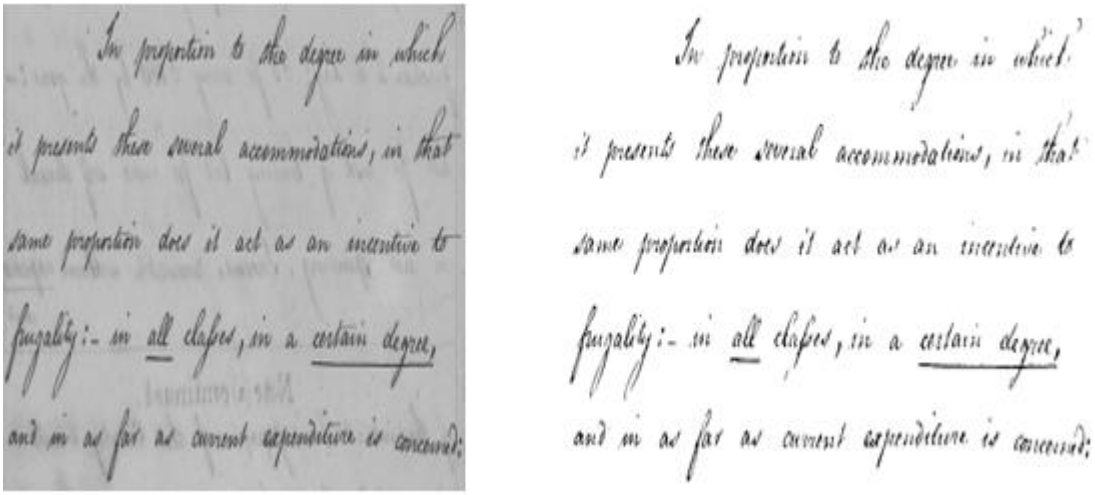

Figure2 Original **[13]** vs Binary Image

2.2.2. Grayscale Representation:

A grayscale image represents colors as various shades of gray, typically ranging from **black** to **white**, with intermediate levels of brightness. Each pixel may have a value representing the intensity of light, typically coded in 8 bits (byte) (256 levels of gray). Grayscale images have served as the foundation for many image processing and computer vision techniques.

Figure3 Original (left) vs grayscale (right) image **[14]**

2.2.3. Color Representation:

A color image is represented by multiple color channels. The most common color models include **RGB** (Red, Green, Blue) and **CMYK** (Cyan, Magenta, Yellow, Black). In the RGB model, each pixel is represented by three components (Red, Green, and Blue), each having a certain intensity, typically 8 bits (byte) per component, which allows for a wide range of colors. Color images are widely used in photographs, videos, and graphics.

Figure4 Original RGB image **[15]**

Figure5 Extracted Red Layer from Original RGB Image **[15]**

Figure6 Extracted Green Layer from Original RGB Image **[15]**

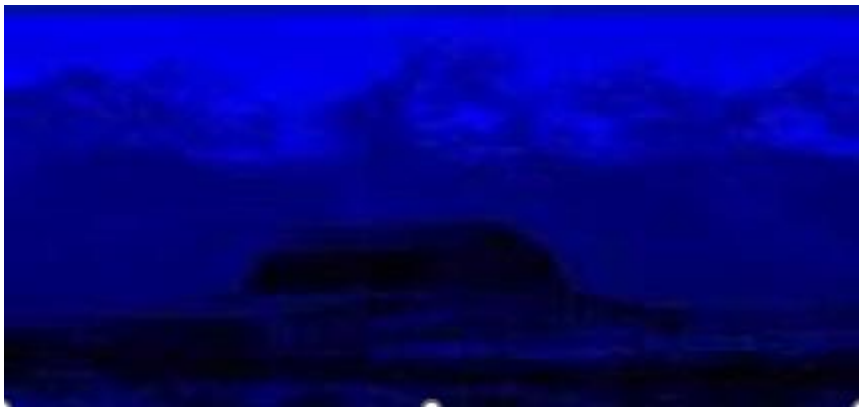

Figure7 Extracted Blue Layer from Original RGB Image **[15]**

2.3 Image Features

Image features are essential attributes extracted from images to characterize it for discrimination purposes. The following are detailed explanations of common types of image features.

2.3.1 Texture

Texture refers to the spatial arrangement of intensity variations in an image. It reflects visual properties such as variability, roughness, and frequency. **[16]**

- **Variability:** measures the differences between pixel values. High variability indicates a rich texture (e.g., tree bark), while low variability suggests a uniform texture (e.g., clear sky). **[16]**
- **Roughness:** describes the irregularity of pixel intensity variations. A rough texture is irregular, whereas a smooth texture is regular. **[16]**
- **Frequency:** Indicates the density of intensity changes. A fine texture has a high frequency, while a coarse texture has a low frequency. **.[16]**

Figure8 Example of Wood Texture **[17]**

Figure9 Example of Sand Texture **[18]**

2.3.1.1 texture techniques

To extract texture features from images, different methods can be applied, including:

a. Local Binary Pattern (LBP)

LBP is a method for describing textures by comparing each pixel with its immediate neighbors in a grayscale image to generate a local binary pattern. These patterns are then represented as a histogram of image matrix, which serves to characterize textured regions.**[19]**

Basic formula of LBP:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, otherwise. \end{cases}$$

(2.1)

where gc is the intensity of the central pixel, and gp are the intensities of its P neighboring pixels.

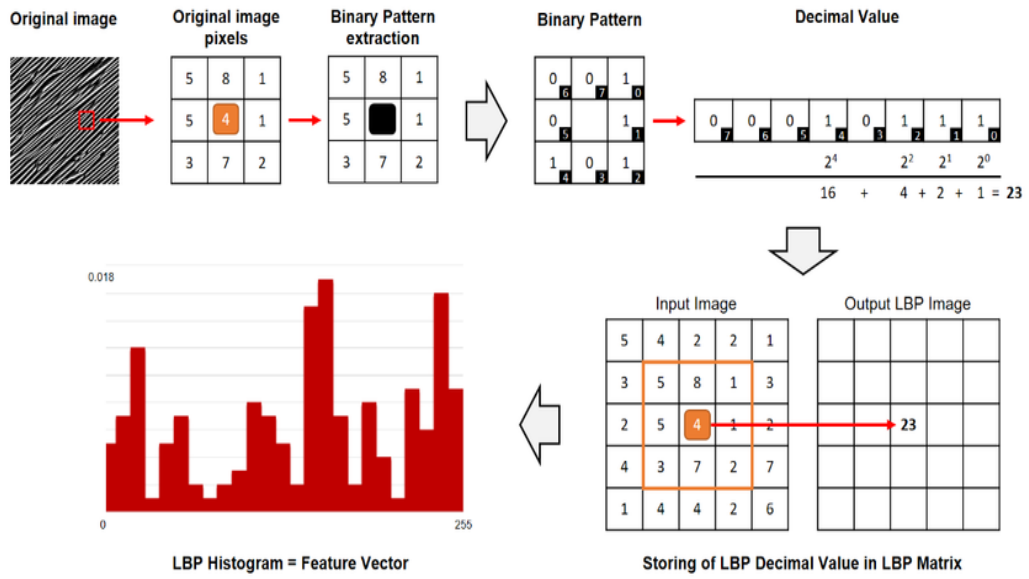

Figure10 Illustration of the LBP matrix and the corresponding histogram **[20]**

b. Co-occurrence Matrix of Haralick:[21]

The Haralick matrix, based on the Gray-Level Co-occurrence Matrix (GLCM), is a statistical representation used to characterize the texture of an image. It extracts various texture features such as energy, entropy, contrast, and correlation from the GLCM. These Haralick features are often used as input attributes for image classification tasks. A classifier can then be trained on these features to predict the class of an unknown image.

It enables the extraction of several texture features such as**:**

- **Energy:**

  Energy measures the uniformity of the image texture.
  The energy of the Haralick co-occurrence matrix is defined as the sum of the squares of the normalized elements of the matrix **M**.

$$E = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \left(\frac{M_{ij}}{N}\right)^2 | \tag{2.2}$$

E: is the energy.

Mij: is the value of the element (i, j) of the co-occurrence matrix **M**.

Ng: is the number of gray levels in the image.

- **Entropy:**

  Entropy measures the complexity of the image texture.
  The entropy of the Haralick co-occurrence matrix is defined as the sum of the products of the matrix elements and their base-2 logarithms.

$$H = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \left(\frac{M_{ij}}{N}\right) \cdot \log_2 \left(\frac{M_{ij}}{N}\right) \tag{2.3}$$

H: is the entropy.

Mij: is the value of the element (i, j) of the co-occurrence matrix **M**.

Ng: is the number of gray levels in the image.

N: is the total number of pixels in the image.

- **Contrast:**

  Contrast measures the difference between neighboring gray levels in the image. The contrast of the Haralick co-occurrence matrix is defined as the sum of the products of the matrix elements and the squared differences of their gray levels.

$$C = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-j)^2 \cdot \left(\frac{M_{ij}}{N}\right) \quad \textbf{(2.4)}$$

C: is the contrast:

Mij: is the value of the element (i, j) of the co-occurrence matrix **M**.

Ng: is the number of gray levels in the image.

N: is the total number of pixels in the image.

- Correlation:

  Correlation measures the dependency between neighboring gray levels in the image. The correlation of the Haralick co-occurrence matrix is defined as the sum of the products of the matrix elements and their normalized mean deviations.

$$\mathbf{C} = \frac{\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} ((i-\mu_i)(j-\mu_j) \cdot \left(\frac{M_{ij}}{N}\right))}{\sigma_i \sigma_j} \quad \textbf{(2.5)}$$

C: is the correlation.

Mij: is the value of the element (i, j) of the co-occurrence matrix **M**.

Ng: is the number of gray levels in the image.

N: is the total number of pixels in the image.

μi: is the mean of the gray levels in row *i* of the image.

μj: is the mean of the gray levels in column *j* of the image.

σi: is the standard deviation of the gray levels in row *i* of the image.

σj: is the standard deviation of the gray levels in column *j* of the image.

2.3.2 Color

Color is one of the most important visual features used to characterize an image. It helps in identifying objects and regions within an image. Color can be represented numerically in various color spaces, such as RGB, HSV, and CIE XYZ, each of which

provides a different way of representing the color information based on the needs of the specific image processing application. **[12]**

2.3.2.1 Color Spaces

A **color space** is a mathematical model used to represent colors. The most commonly used space is **RGB (Red, Green, Blue)**, which is based on the additive combination of the three primary colors.

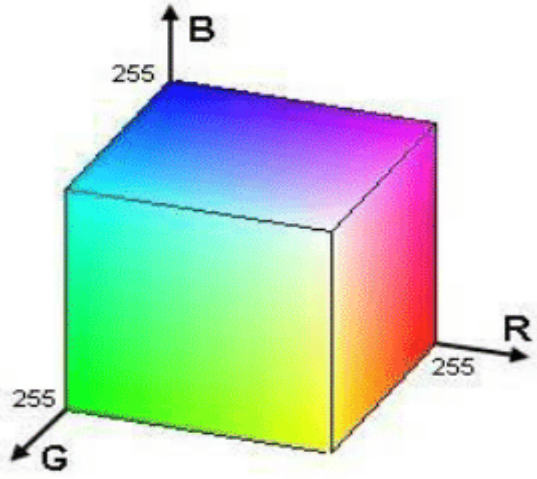

Figure11 RGB model representation [**23]**

Other color spaces are often preferred for specific tasks:

- HSV **(Hue, Saturation, Value)** separates chromatic content from intensity and is commonly used in object recognition due to its alignment with human visual perception.**[22]**

Figure12 HSV model representation **[24]**

- **LAB** (or CIELAB), defined by the International Commission on Illumination (CIE), is perceptually uniform, meaning color differences correspond to human-perceived differences.**[22]**

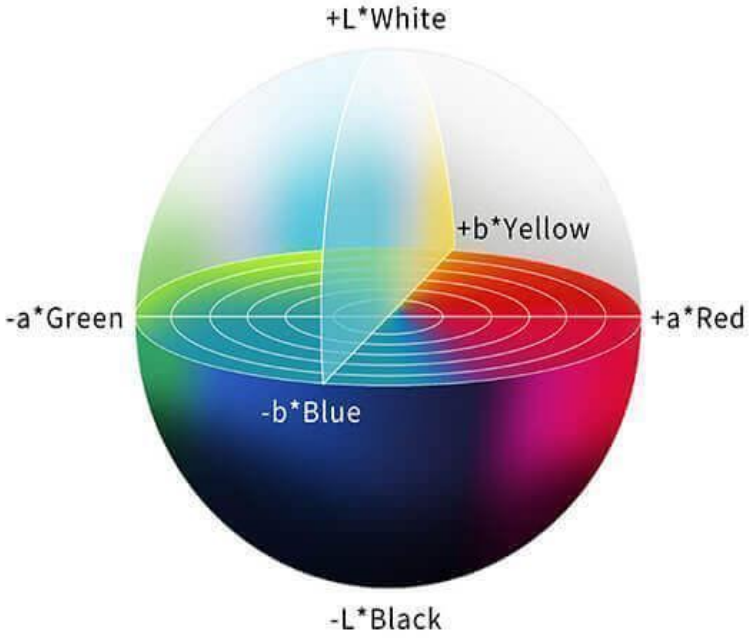

Figure13 LAB model representation **[25]**

- **YCbCr** is widely used in video compression applications, as it separates luminance (Y) from chrominance components (Cb and Cr), which is useful for reducing redundancy in color data. .**[22]**

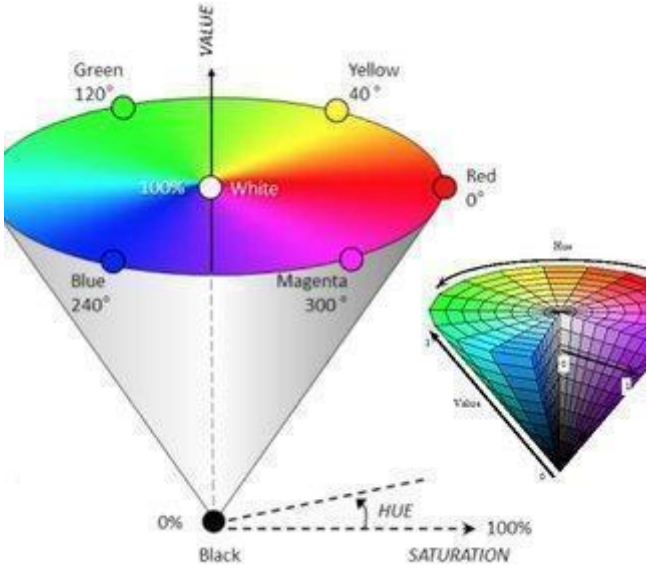

Figure14 YCbCr model representation [**26]**

Selecting an appropriate color space can enhance the robustness of feature extraction, especially in the presence of illumination changes or geometric transformations.

2.3.2.2 Color Histogram

The **color histogram** is a statistical representation that describes the **distribution of colors** within an image. It counts the number of pixels corresponding to each color level within a specific channel (e.g., R, G, and B in RGB space).**[27]**

Here is the equation of the color histogram for a given color channel ccc of an image I:

$$h(k) = N(I, c = k) \quad \textbf{(2.6)}$$

h(k): is the number of pixels in image I that have a color value k for the channel c.

N (I, c = k): is the number of pixels in image I for which the value of channel c is equal to k.

For instance, two images containing similar objects but with different dominant colors can be effectively distinguished using their respective color histograms.

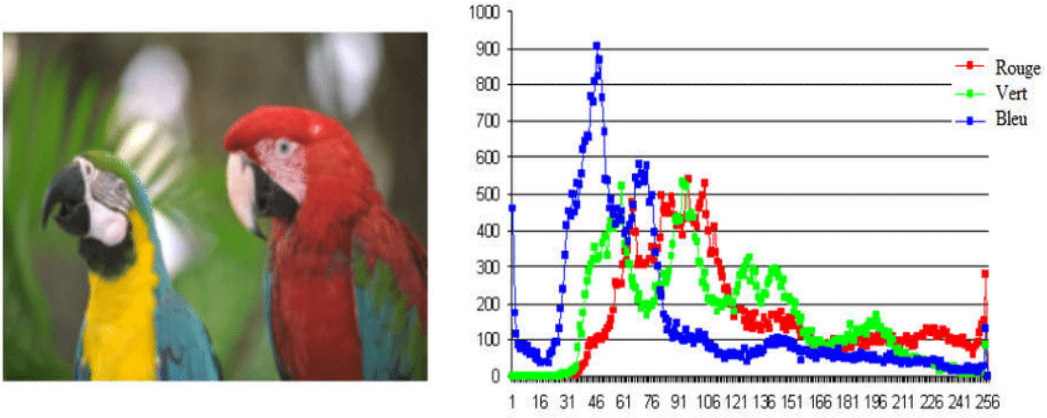

Figure15 Image and corresponding histograms for red, green, and blue channels **[28]**

2.3.2.3 Color Moments

Color moments are statistical measures used to capture the distribution of color in an image. They are based on the assumption that the distribution of color values in an image can be interpreted as a probability distribution. [**29**] The most commonly used moments are: **[30]**

- Mean: Mean can be understood as the average color value in the image.

$$E_i = \sum_{j=1}^{N} \frac{1}{N} p_{ij} \quad \textbf{(2.7)}$$

- where N is the number of pixels in the image and Pij is the value of the j-th pixel of the image at the i-th color channel.

- Standard Deviation:

  The standard deviation is the square root of the variance of the distribution.

$$\sigma_i = \sqrt{(\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^2)} \quad \textbf{(2.8)}$$

  where Ei is the mean value, or first color moment, for the i-th color channel of the image.

- Skewness:

  Skewness can be understood as a measure of the degree of asymmetry in the distributi on.

$$s_i = \sqrt[3]{(\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^3)} \quad \textbf{(2.9)}$$

2.3.3 Shape

Shape descriptors provide valuable information about the geometry and structural patterns of objects within an image.[**31]** Among the most powerful tools for representing shape are Hu's moments, Legendre polynomials, and Fourier descriptors.

2.3.3.1 Hu's Invariant Moments

Hu moments are a group of seven descriptors derived from image moments that are invariant to translation, rotation, and scale. These properties make them highly suitable for recognizing shapes in varying conditions.[**31]**

Key advantages include:

- **Low-dimensional representation:** With only seven values, they significantly reduce the data size while preserving essential shape characteristics. .[**31]**

- **Robustness to transformations:** They enable consistent shape comparison even when images are rotated or resized.[**31]**
- **Efficiency in classification:** Hu moments have proven effective in improving classification performance by encoding geometrical information compactly.[**31]**

2.3.3.2 Legendre Moments

Legendre moments are derived from orthogonal Legendre polynomials and are used to represent the global shape of an object. They offer mathematical compactness and are often used for capturing smooth variations in object contours. **[31]**

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \int_{-1}^{1} \int_{-1}^{1} P_p(x) P_q(y) f(x,y)\, dx\, dy \tag{2.10}$$

where Pp(x) and Pq(y) are the Legendre polynomials of degree p and q, respectively. These moments provide a compact representation of image features.

Benefits of using Legendre moments include:

- Compact global shape encoding: Useful for differentiating objects based on overall form.**[31]**
- Scale and rotation invariance: Facilitates comparison across transformed images**.[32]**

2.3.3.3 Fourier Descriptors

Fourier descriptors analyze the frequency content of object contours by transforming them into the frequency domain. They are especially useful for capturing detailed structural features.**[33]**

$$F(k_1, k_2) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-\frac{1}{2\pi} j(k_1 i + k_2 j)} \tag{2.11}$$

Where:

- F (k1, k2) are the Fourier coefficients at frequencies (k1, k2),
- f (i, j) is the image function at pixel coordinates (i, j),

- k1 and k2 are the frequency indices

- N is the size of the image

Advantages:

- Analysis of boundary complexity: High-frequency components correspond to intricate edges.[**33]**
- Invariance to translation and rotation: Ensures consistent shape representation.[**33]**
- Complementarity: When combined with global descriptors (like Legendre moments), they provide a richer shape representation**.[33]**

D. Applications and Strengths

These descriptors are employed in various domains:

- Biometric identification: Facial and iris recognition systems.
- Defect detection: In industrial inspection and quality control.
- General object recognition: In logos, handwritten digits, and medical shapes.

2.4 Distance and Similarity Measures

Distance measures and similarity metrics are crucial for quantifying the relationship between data points or image features. The following sections provide detailed explanations of common distance and similarity measures.

2.4.1 Distance Measures

Distance measures are functions that quantify the dissimilarity between two objects, often represented as vectors in a multidimensional space. They are fundamental for evaluating how much two objects differ, taking into account their respective features.**[34]**

In the context of image processing, consider two images represented by their corresponding feature vectors X= (x1, x2,.., xn) and Y= (y1, y2,…,yn).

Among the most commonly used measures are:

2.4.1.1 Euclidean distance:

Euclidean distance is the length of the straight-line segment connecting two points in Euclidean space. In such a space, the distance formulas for points in rectangular coordinates are based on the Pythagorean theorem.

The Euclidean distance between two images X and Y is given by:

$$d(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.12}$$

2.4.1.2 Manhattan distance: Manhattan distance, also known as taxicab distance or city block distance, is the sum of the absolute differences of their coordinates. **[35]** the Manhattan distance is given by:

$$d(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.13}$$

2.4.1.3 Canberra distance: The Canberra distance is a numerical measure of the distance between pairs of points in a vector space, introduced in 1966 and refined in 1967 by G. N. Lance and W. T. Williams. It is a weighted version of L_1 (Manhattan) distance**.[36]**

Canberra distance $$d_c(X,Y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{x_i + y_i} \tag{2.14}$$

2.4.2 Similarity measures

In statistics and related fields, a similarity measure is a real-valued function that quantifies the similarity between two objects in the N-dimensional features space**.[37]**

2.4.2.1 correlation factor

A correlation coefficient is a numerical measure of some type of linear correlation, meaning a statistical relationship between two variables.**[37]**

$$\operatorname{corr}(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{2.15}$$

The **correlation coefficient** is represented as follows:

xi and yi are the individual values of the two variables,

- .$\underline{x}$ and $\underline{y}$ are the means of the values of x and y, respectively.

2.4.2.2 Cosine similarity

Cosine similarity is a measure of similarity defined as the cosine of the angle between two vectors. It is often used to compare images that have been represented as feature vectors.[**38]** it is widely applied in **Information Retrieval** to compare documents. The formula is as follows:

$$\cos(\mathbf{X}, \mathbf{Y}) = \frac{\mathbf{X} \cdot \mathbf{Y}}{\|\mathbf{X}\| \cdot \|\mathbf{Y}\|} = \cos(\theta) \tag{2.16}$$

More explicitly, this becomes:

$$\cos(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.17}$$

where θ is the angle between the two vectors.

- When $\cos(\theta) \approx 1$, the angle $\theta \approx 0^\circ$: the vectors are very similar.
- When $\cos(\theta) \approx 0$, the angle $\theta \approx 90^\circ$: the vectors are orthogonal (no similarity).
- When $\cos(\theta) \approx -1$, the angle $\theta \approx 180^\circ$: the vectors are opposite (completely dissimilar).

3. Main Image Mining Tasks

This section includes some of the main tasks involved in image mining.

3.1 Image Classification

Image classification is a core task in image mining that involves assigning a semantic label to an image based on its visual content. It plays a critical role in understanding, organizing, and retrieving information from large image datasets.

In this context, a set of labeled images is used to train a machine learning model to recognize patterns and assign class labels to new, unseen images. This approach relies on predefined categories and leverages learning algorithms to infer the decision boundaries between them.**[39]**

3.2 Image Clustering

Image clustering is an unsupervised image mining task that aims to group unlabeled images into meaningful clusters based on visual similarity without relying on predefined class labels. It operates by analyzing low-level features such as color, texture, and shape to

discover inherent patterns in image datasets. Various techniques such as k-means, hierarchical clustering are used to detect groups of visually similar images. For example, in 2000, the ACQ (Automatic Clustering and Query) method was proposed, applying wavelet transforms to efficiently detect clusters of arbitrary shapes in high-dimensional feature spaces. In 2001, clustering methods were used to visualize typhoon cloud patterns, while in 1999, clustering was employed to identify unauthorized image copying online. Overall, image clustering enhances image storage, indexing, and retrieval, making it an essential component of intelligent image mining systems. **[40]**

3.3 Anomaly Detection

Anomaly detection is a key data analysis task that aims to identify outliers that deviate significantly from what is considered normal. These anomalies may reflect critical issues such as system errors, security breaches, medical conditions, or abnormal behaviors.

The effectiveness of an anomaly detection system depends on the quality of the data, the nature of the anomalies, and the context in which it is applied. In practical applications, such as in medical imaging, anomaly detection plays a supporting role in highlighting irregularities. For instance, in magnetic resonance imaging (MRI), unsupervised learning methods can be used to detect unexpected tissue patterns without needing pre-labeled data, aiding in the identification of potential pathologies.

While anomaly detection techniques offer promising results, challenges persist such as algorithmic limitations, interpretability issues, and implementation costs. To address these concerns, modern research explores lightweight, accessible, and interpretable solutions that align with goals such as early detection, cost-effectiveness, and scalability particularly in sensitive fields like healthcare and cybersecurity.**[41]**

3.4 Content Based Image Retrieval (CBIR)

Content-Based Image Retrieval (CBIR) is an image mining technique used to automatically search and retrieve images from large databases based on their visual content, such as color, texture, shape, or other features, without relying on manual annotations. CBIR has wide applications across various fields including medicine, security, design, geography, archiving, and retail.

The CBIR process mainly involves two key steps: feature extraction from images, and similarity-based retrieval using suitable distance or similarity measures. Due to the rapid growth in the size of image datasets, developing efficient methods to speed up the retrieval

phase has become a critical research focus. Recent advancements incorporate machine learning, distributed processing (e.g., GPU acceleration), and intelligent indexing structures to improve both the performance and efficiency of CBIR systems.

Modern CBIR systems often combine multiple visual descriptors, such as color moments, local binary patterns (LBP), and may utilize supervised or unsupervised learning approaches to enhance retrieval accuracy. The ultimate goal is to deliver relevant search results quickly and accurately, especially in large-scale image repositories.**[42]**

3.5 Motifs and association rules discovery

Motifs and association rules discovery mining are fundamental tasks in data mining, aimed at extracting significant and recurrent patterns from large datasets. A *motif* refers to a combination of elements (also called *items*) that appears frequently, while an *association rule* is an implication of the form $A \rightarrow B$, indicating that the presence of item A in a transaction often implies the presence of item B.

These techniques are widely applied in various domains such as market basket analysis, bioinformatics, recommendation systems, and fraud detection. Association rules are typically evaluated using statistical measures such as support, confidence which help determine their frequency within the data.

Application in Image Mining

The goal is to discover hidden relationships among **visual features extracted from images**. Each image can be transformed into a symbolic transaction by encoding its features (such as color, texture, or shape) into discrete visual items.**[42]**

Benefits and Limitations

The use of motif discovery and association rules in image mining offers several advantages:

- **Interpretability**: The extracted rules are human-readable and explainable.
- **Unsupervised Learning**: No prior labeling of images is required.

However, some limitations exist:

- **Loss of spatial information** during feature quantization.
- **High rule redundancy**, which can make interpretation difficult.
- **Computational complexity**, especially with large-scale image datasets.

4. Conclusion

In summary, this chapter has provided a comprehensive introduction to image mining, focusing on the fundamental concepts and techniques. We began by exploring the basic principles of images and the extraction of important image features. These elements are essential for understanding how images are processed and analyzed within the context of image mining. Furthermore, we examined the main tasks of image mining. By highlighting these core concepts and tasks, this chapter has led the groundwork for a deeper understanding of image mining techniques and their applications.

Chapter 3: Image Classification

1. Introduction

In this chapter, we delve into the concept of Image Classification, an essential task in the realm of image mining. Image classification refers to the process of categorizing images into predefined classes, based on their visual content. This task has gained tremendous importance in various fields, due to its ability to automate the interpretation of visual data.**[43]**

We begin by outlining the process of image classification, which involves several crucial steps to transform raw image data into meaningful classifications. These steps include data preprocessing, image characterization, selection of classifiers, model training, model testing, and evaluating its performance using various metrics.

The chapter further explores a range of machine learning techniques commonly applied to image classification. These techniques include Support Vector Machines (SVMs), Naive Bayes Classifier, K-Nearest Neighbors (K-NN), Decision Trees, and Random Forests. Each of these algorithms has its strengths and limitations, which are examined in the context of image classification tasks.**[44]**

While machine learning techniques have proven successful, they also face challenges, such as overfitting, scalability issues, and the need for handcrafted features.**[45]**

These limitations are addressed in the chapter, and we also introduce deep learning techniques, such as Convolutional Neural Networks (CNNs), which have revolutionized image classification by automatically learning complex features from raw data.**[43]**

2.Definition of Image Classification

Image classification is the process of assigning one or more labels to a digital image based on its visual content. This process is part of the field of image data mining and allows images to be categorized according to the information they contain. It can be performed in a supervised manner, where a model is trained with labeled image examples, or in an unsupervised manner, where the algorithm identifies structures or patterns in the data without prior human intervention.

3.Process of Image Classification

Image classification is a critical task in the field of image mining, involving the categorization of an image into predefined classes or categories. The process typically

involves several key steps, each contributing to the extraction of meaningful information from the raw image data.

3.1 Data Preprocessing

Data preprocessing is a crucial step in the image classification pipeline, as it prepares raw image data for more effective analysis by machine learning models. The goal is to enhance image quality and ensure consistency across the dataset, which can significantly improve classification accuracy. Common preprocessing techniques include resizing images to a uniform dimension, cropping to focus on regions of interest, normalizing pixel values to standardize input ranges, and adjusting lighting or contrast to reduce the effects of poor illumination. These operations help reduce noise and variability, allowing the classifier to learn more robustly. **[46]**

3.2 Image characterization (Feature Extraction)

Image characterization involves the extraction of relevant visual features from preprocessed images to represent their key information for classification tasks. These features capture essential aspects of an image, such as color, texture, shape, and edges, which are vital for distinguishing between different classes. Color features may include color histograms or color moments, while texture features could be derived from statistical measures or filters like Gabor or wavelet transforms. Shape features typically represent the structural patterns of objects within the image, and edge features focus on the boundaries and contours of objects. By extracting and quantifying these characteristics, we can create a compact, informative representation of each image. **[47]**

3.3 Selection of Classifier

The selection of an appropriate classifier is a crucial step in the image classification process, as it determines the model that will be used to categorize images into predefined classes. Various classifiers are available, each with its own strengths and weaknesses depending on the nature of the data and the task at hand. Commonly used classifiers include support vector machines (SVM), k-nearest neighbors (k-NN), decision trees, random forests, and neural networks. The choice of classifier is influenced by several factors, such as the complexity of the feature space, the size of the dataset, and the computational resources available. For example, SVM is often preferred for its ability to perform well in high-dimensional spaces, while decision trees

are useful for interpretability. Neural networks, particularly deep learning models, are increasingly used for large-scale image classification tasks due to their ability to automatically learn hierarchical features from raw data. The effectiveness of the classifier can be further enhanced by tuning its hyperparameters and using ensemble methods to combine multiple models for improved performance.

3.4 Model Training

Model training is a fundamental step in the image classification process, where the classifier learns to associate extracted image features with their corresponding labels, using a labeled dataset where each image is paired with a known class or category. During the training phase, the model adjusts its internal parameters to minimize the error between its predictions and the actual labels. This process often involves the use of optimization algorithms such as gradient descent to fine-tune the model's weights. The size and quality of the training dataset are crucial, as the model needs a diverse and representative set of images to generalize well to unseen data. Overfitting and underfitting are common challenges during training, and techniques such as cross-validation, regularization, and data augmentation are often employed to address these issues and ensure the model’s robustness.

3.5 Model Testing

Model testing is the phase where the trained classifier is evaluated on new, unseen images to assess its generalization ability. During this stage, the visual features of the new images are extracted and compared to the features learned from the training dataset. The model then assigns the most probable labels to the new images based on these comparisons. It is important to test the model on a diverse set of images. This helps to ensure that the model is not overfitting to the training data and can generalize well to new examples. If the results are not satisfactory, further improvements can be made through techniques like hyperparameter tuning, adding more training data, or using more advanced models.

3.6 Performance Evaluation

Performance evaluation is a critical step in assessing the effectiveness of a classification model. After testing the model on new, unseen images, various metrics are used to

measure its accuracy. Common evaluation metrics for image classification include accuracy, precision, and F1-score, each providing a different perspective on model performance. **[48]**

Accuracy measures the overall proportion of correct predictions and is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Precision assesses the proportion of true positive predictions out of all predicted positive instances:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

Recall measures the proportion of actual positive instances that are correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

F1-score is the harmonic mean of precision and recall, providing a balanced measure between them:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

Where:

TP is the number of true positives,

TN is the number of true negatives,

FP is the number of false positives,

FN is the number of false negatives.

In addition, confusion matrices are often used to visualize the performance of a classifier, showing the counts of true positives, true negatives, false positives, and false negatives. Cross-validation techniques, such as k-fold cross-validation, are also employed to ensure that the model's performance is consistent across different subsets of the data and to mitigate the risk of overfitting. A model with high performance is one that generalizes well to new data while maintaining a low error rate.

4. Machine Learning Techniques

This section presents some of the machine learning techniques used in image classification.

4.1 Support Vector Machine

Support Vector Machines (SVMs) are supervised learning models widely used for image classification. The core principle behind SVMs is to find a hyperplane that best separates data points from different classes in a feature space. The goal is to maximize the margin between classes, which ensures robust classification**.[49]**

Use of Kernels in SVM

SVMs are particularly effective for handling non-linear data through the use of kernels. A kernel is a mathematical function that allows data to be projected into a higher-dimensional space. This projection makes it possible to find linear separations in an otherwise non-linearly separable data space. The kernel facilitates classification by making the decision boundaries between classes more distinguishable. The kernel facilitates classification by making the decision boundaries between classes more distinguishable.

There are several types of kernels, each having specific advantages depending on the nature of the data. Some commonly used kernels include:

- **Linear Kernel:** Used when the data is already linearly separable in the original space.
- **Polynomial Kernel**: Applied for more complex relationships between data points.
- **Radial Basis Function (RBF) Kernel:** Particularly popular, it is used when data is non-linear and allows for a flexible transformation of data into a higher-dimensional space.

Support Vector Machines (SVMs) are widely used in various image classification tasks, including:

- **Face Recognition:** Identifying individuals based on facial features.
- **Medical Image Analysis:** Detecting diseases or abnormalities in medical scans like MRIs and X-rays.
- **Image Retrieval**: Matching query images to a database of labeled images in content-based retrieval systems.

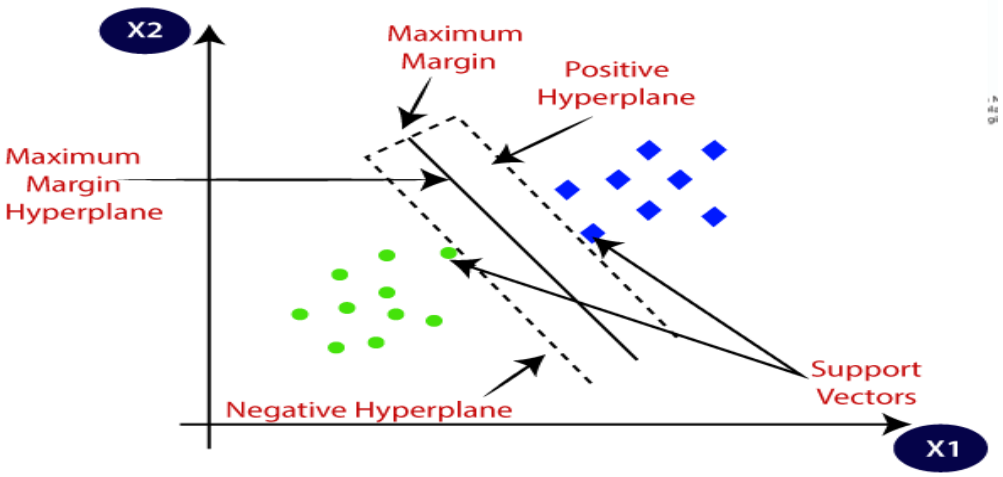

Figure16 Support Vector Machine (SVM) illustration [**50]**

4.2 Naïve Bayes Classifier

Naïve Bayes is a type of probabilistic classifier based on the famous Bayes theorem **[51], which** applies the principle of conditional probability. It is widely used for classification tasks, especially when the input features are independent (which is why it is called naïve **[52].** Despite this assumption of feature independence, Naive Bayes can perform surprisingly well even in real-world scenarios where features are not fully independent. **[53]**

Core Concept

Naive Bayes works by calculating the probability of each class given the features of the data. The algorithm makes an assumption that the features are conditionally independent given the class label. Using Bayes' Theorem, it computes the posterior probability of a class, which is given by:

$$P(A|B) = P(B|A).\ P(A)/P(B) \quad \textbf{(3.5)}$$

Where:

- P(A|B) is the probability of class A given the features B.
- P(B|A) is the likelihood of the features B given the class A.
- P(A) is the prior probability of class A.
- P(B) is the total probability of the features B.

Steps in Naive Bayes Classification

1. **Training Phase:**
    - Calculate the prior probability P(A) for each class in the dataset.

- Calculate the likelihood P(B|A) for each feature given the class (typically using statistical measures such as mean and variance for continuous data, or frequency for categorical data).

2. **Classification Phase:**
   - For a new instance, calculate the posterior probability for each class using the learned parameters.
   - Assign the class label with the highest posterior probability.

Advantages

- Simple and fast: Naive Bayes is easy to implement and computationally efficient. **[54]**
- Effective with categorical features: Particularly good for text classification and spam detection. **[54]**
- Performs well with independent features.**[55]**

Disadvantages

- Independence assumption: The assumption that features are independent is often unrealistic in real-world problems, leading to suboptimal performance.

4.3 K-Nearest Neighbor

The k-Nearest Neighbors (KNN) algorithm is one of the simplest and most widely used techniques in data mining, based on the principle of learning by similarity. It is a supervised learning method. The core idea of KNN is to classify a new instance by considering the k closest training examples in an n-dimensional feature space, using a similarity measure such as Euclidean distance. Unlike other machine learning models, KNN does not construct an explicit classification model during training; instead, it stores the entire dataset and makes predictions based on the majority class of the nearest neighbors. This non-parametric and instance-based learning approach makes KNN highly flexible. **[56]** However, it can be computationally expensive for large datasets due to the need to compute distances for every new instance. KNN is widely used in various domains including pattern recognition, recommendation systems and medical diagnosis.

The foundational steps involved in k-NN algorithm are as follows**: [57]**

1. Distance calculation
2. Sorting dictionary

3. Voting with lowest k distances

Pseudocode is given as:

For every point in our dataset:

– calculate the distance between inX and the current point.

– sort the distances in increasing order.

– take k items with lowest distances to inX.

– find the majority class among these items.

– return the majority class as our prediction for the class of inX.

inX: the input point to classify.

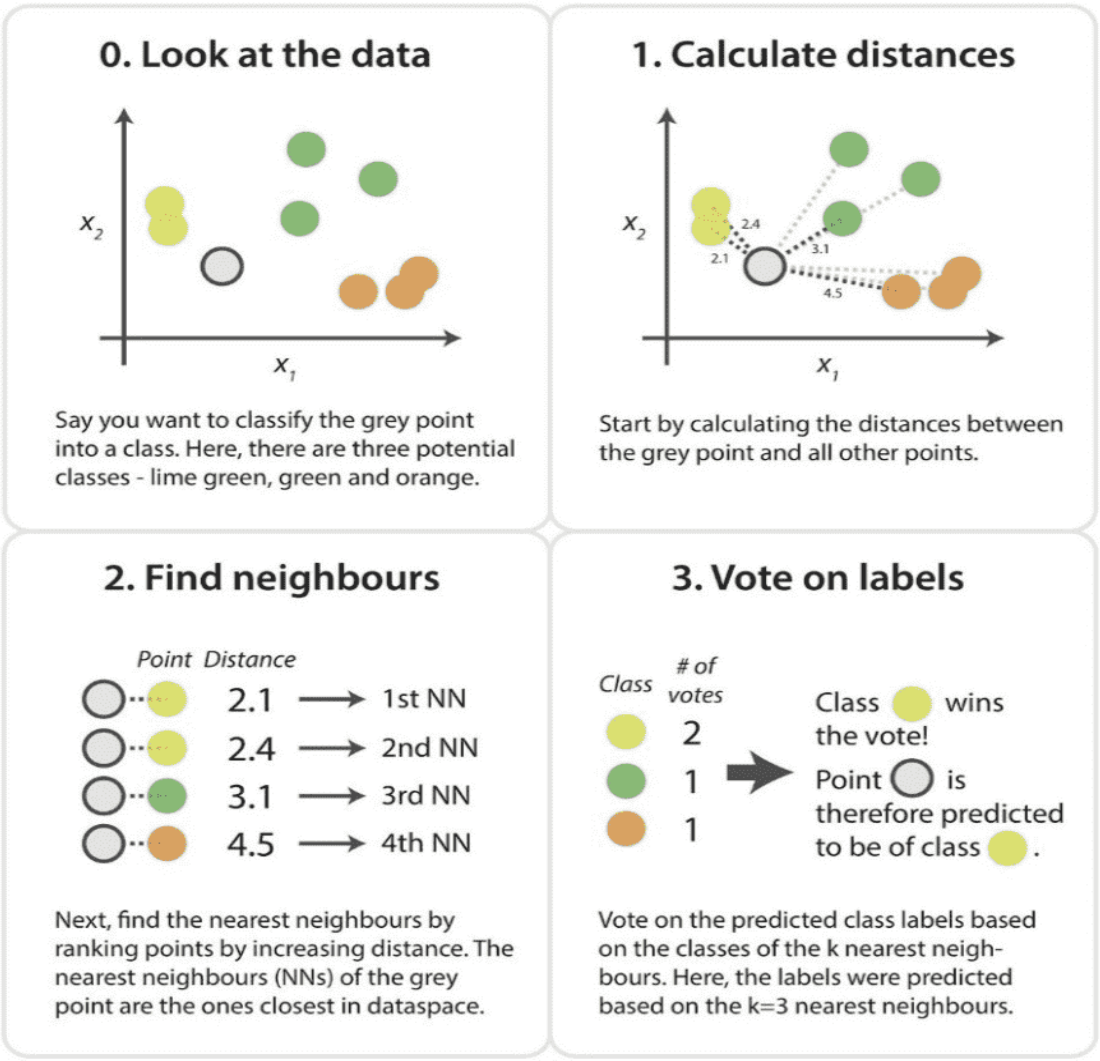

Figure17 K-Nearest Neighbors (KNN) algorithm **[58]**

4.4 Decision Trees

Decision trees are widely used in data mining for classification and regression due to their simplicity, speed and interpretability **[59].** A decision tree follows a hierarchical structure of logical rules, dividing a dataset into subsets based on the most discriminative attributes at each level. Several algorithms exist, such as ID3, which uses entropy and information gain, C4.5, which improves ID3 by handling continuous and missing data, CART, which generates binary trees based on the Gini index for classification and mean squared error for regression, and CHAID, which applies the Chi-square test and allows multiway splits. However, they can be sensitive to noise, prone to overfitting if too complex, and less efficient on large datasets **[60].** Despite these challenges, they are widely applied in fields such as finance (credit risk assessment), healthcare (medical

diagnosis), marketing (customer segmentation), and computer vision (image recognition).

The ID3 algorithm

Pseudocode: [61]

This pseudocode assumes that the attributes are discrete and that the classifications are either yes or no. It deals with inconsistent training data by choosing the most popular classification label whenever a possible conflict arises.

```
defid3(examples, classification_attribute, attributes):
    create a root node for the tree
    if all examples are positive/yes:
        return root node with positive/yes label
    else if all examples are negative/no:
        return root node with negative/no label
    else if there are no attributes left:
        return root node with most popular classification_attribute label
    else:
        best_attribute = attribute from attributes that best classifies
examples
        assignbest_attribute to root node
        for each value in best_attribute:
            add branch below root node for the value
            branch_examples = [examples that have that value for
best_attribute]
            if branch_examples is empty:
                add leaf node with most popular classification_attribute label
            else:
                add subtree id3(branch_examples, classification_attribute,
                                                    attributes -
best_attribute)
```

calculate it using the formula for entropy, which is:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i \qquad \textbf{(3.6)}$$

In this formula, c corresponds to the number of different classifications and pi corresponds to the proportion of the data with the classification i. Because our example -

Information gain measures the duction in entropy that results from partitioning the data on an attribute A, which is another way of saying that it represents how effective an attribute is at classifying the data. Given a set of training data S and an attribute A, the formula for information gain is:

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v) \quad \textbf{(3.7)}$$

Where:

S: the total set of examples (the dataset);

A: the attribute for which the gain is being calculated;

Values(A): the set of all possible values that attribute A can take;

Sv: the subset of S where attribute A has the value v;

|Sv|/|S|: the proportion of examples in S for which A=v;

Entropy (Sv): the entropy of the subset Sv.

The entropies of the partitions, when summed and weighted, can be compared to the entropy of the entire data set. The first term corresponds to the entropy of the data before the partitioning, where as the second term corresponds to the entropy after wards. We want to maximize information gain, so we want the entropies of the partitioned data to be as low as possible, which explains why attributes that exhibit high information gain split training data into relatively heterogeneous groups.[**60**]

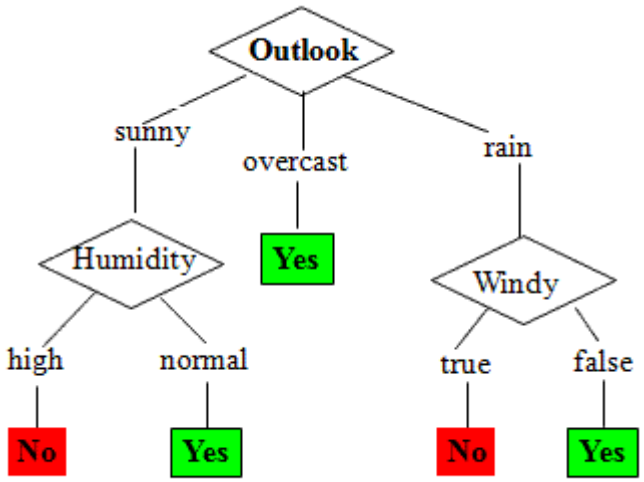

Figure18 An illustrative example of a decision tree **[61]**

4.5 Random Forests

Random Forests are a supervised learning algorithm that belongs to the family of ensemble methods. It operates by constructing and aggregating a large number of decision trees, each trained independently on a randomly sampled subset of the original dataset using bootstrap sampling. This process introduces diversity among the individual trees, which helps reduce overfitting and increases prediction robustness. The final prediction is obtained by combining the outputs of all trees, resulting in a model that is generally more stable and accurate than a single decision tree. [**62**]

Two fundamental mechanisms ensure the independence of the trees in a Random Forest:

- Bootstrapping (bootstrap aggregating or bagging): From a training set containing *n* instances, a new sample of size *n* is generated for each tree through random sampling with replacement. As a result, some examples appear multiple times in the sample, while approximately one-third of the data are not selected at all. This process creates training sets that are similar yet distinct, which introduces variability among the trees**. [63]**
- Random selection of predictors at each split: Unlike a traditional decision tree that considers all available variables to determine the best split at each node, Random Forest randomly restricts this choice to a subset of variables. This constraint promotes diversity among the trees in the forest**.[64]**

During the prediction phase, each tree votes for a target class. The model's final prediction is determined through majority voting: the class receiving the most votes among all the trees is chosen as the model's output. **[65]** For example, if 800 out of 1000 trees predict that a patient belongs to tumor subtype A, then the final prediction will correspond to that subtype.

The main advantage of Random Forest lies in its ability to reduce the variance characteristic of decision trees. Although decision trees are capable of capturing complex data structures, their flexibility makes them sensitive to variations in the training set, resulting in high variance. Random Forest mitigates this instability by averaging the predictions of multiple independent trees, thereby producing a more robust and generally more accurate model. **[64]**

5. Limitation of Hand-Crafted Machine Learning methods

While machine learning has significantly advanced various domains, including image classification, it also presents several limitations:

Dependence on Feature Engineering: ML algorithms, such as support vector machines or decision trees, rely heavily on manually extracted features. The quality of these features has a direct impact on model performance. In complex domains like medical imaging, defining effective features often requires expert knowledge and may not capture all the underlying patterns in the data. This challenge is especially apparent in studies like Nasief et al. (2020), where radiomic features play a crucial role in predicting treatment response for pancreatic cancer**. [65]**

Limited Performance on High-Dimensional Data: ML models often struggle with high-dimensional inputs such as images, where the number of features can be extremely large. This makes the learning process less effective without dimensionality reduction techniques. **[66]**

Generalization Issues: Machine learning models may perform well on training data but fail to generalize to unseen data due to overfitting, especially when the dataset is small or imbalanced**. [67]**

6. Deep Learning Techniques

In the field of image classification, deep learning offers powerful tools capable of automatically learning complex patterns and features from raw image data. Below are the main deep learning techniques specifically used for image-based tasks:

6.1 Convolutional Neural Network (CNN):

Convolutional Neural Networks (CNNs) are among the most widely used and effective deep learning techniques for image classification. CNNs are specifically designed to handle 2D data such as images by automatically extracting important visual features from them. **[68]**

CNN Architecture:

A typical CNN consists of the following key layers:

- **Convolutional Layers**: Apply filters (kernels) across the input image to detect local patterns such as edges, textures, or shapes.

- **Pooling Layers** (usually **Max Pooling**): Reduce the spatial size of the feature maps while preserving key information, thus speeding up computation and reducing overfitting.
- **Fully Connected Layers (FC)**: These layers combine the extracted features and perform the final classification.

Example of How CNN Works (Simplified):

1. The image is passed into the network;
2. Several convolutional layers extract local patterns;
3. Pooling layers reduce feature map sizes;
4. Fully connected layers output the final classification (e.g., "benign tumor" or "malignant tumor").

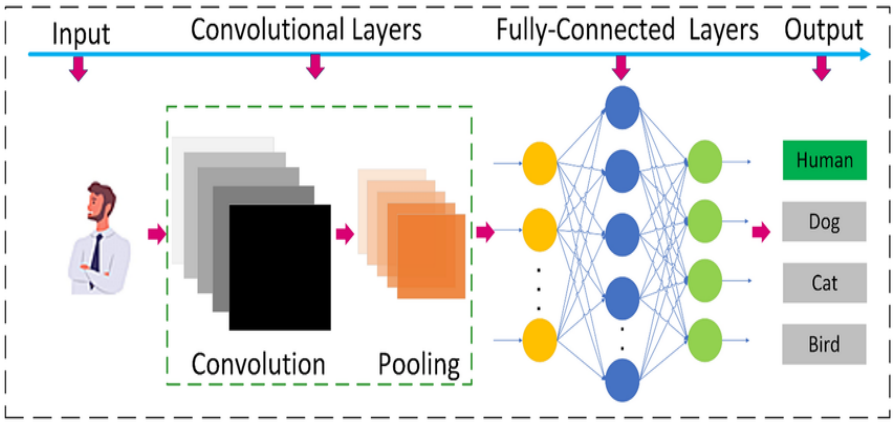

Figure19 Illustration of the internal structure of a CNN used for image classification **[69]**

7. Conclusion

In conclusion, this chapter has provided a detailed overview of image classification, covering its definition, key processes, and various techniques. We explored the steps involved in the image classification process and examined the different machine learning techniques, highlighting their applications and limitations in the context of image classification. Finally, we introduced deep learning techniques, emphasizing their transformative impact on the field.

Chapter 4: Experimental Setup and Implementation

1. Introduction

In this chapter, we focus on the practical aspect of our work through a comparative study involving several classifiers: KNN, SVM, Decision Tree, Naive Bayes, and Random Forest. The study also includes a deep learning-based classifier CNN (Convolutional Neural Network) which leverages what are known as deep features or implicit features.

For the experiments, we used two well-known image datasets: the Corel-1000 dataset (also known as Wang or Corel1K, named after its creator) and the Kimia Path960 dataset.

We begin by presenting the programming environment used for the implementation, followed by a detailed description of the experiments conducted. Then, we present the results in the form of performance graphs, such as accuracy and execution time. Finally, we include a few interface screenshots to demonstrate the evaluation of the CNN model trained on the Corel1K dataset using sample test images.

2. Used Image Datasets

2.1 Corel 1K

The Corel-1K dataset is widely used for image classification tasks. It consists of 1000 images, evenly distributed across 10 semantic categories, each containing 100 images. The categories cover diverse themes such as African people and villages, beaches, buildings, buses, dinosaurs, elephants, flowers, horses, mountains, and food.

The images in the Corel-1K dataset vary in their original sizes and are all color images in the **RGB (Red, Green, Blue)** format.

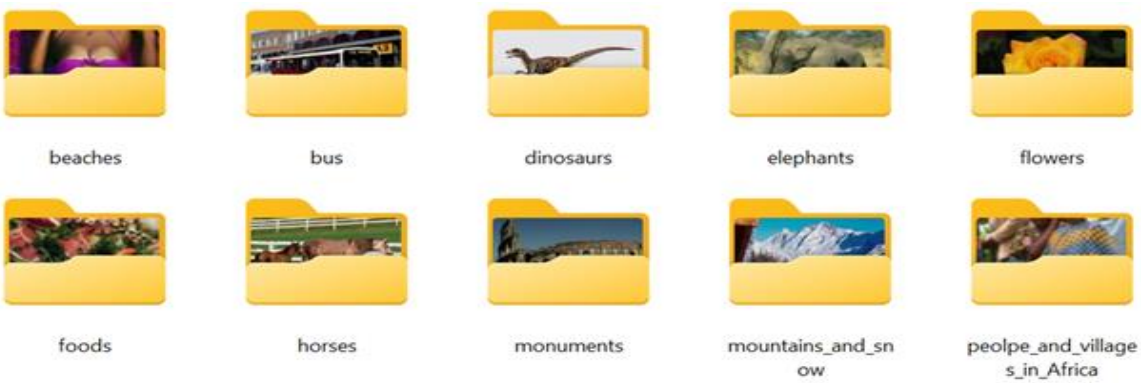

Figure20 The Corel-1K dataset

2.2 KIMIA path 960:

The KIMIA Path960 dataset is a digital pathology image collection consisting of 960 images distributed across 20 classes, with 48 images per class. It includes histopathological image patches extracted from more than 400 whole slide images (WSIs), representing various tissue types such as muscular, epithelial, and connective tissues, among others.

Figure21 The KIMIA Path960 dataset

All images are in color (RGB format) and have a fixed resolution of 308×168 pixels, which facilitates preprocessing. KIMIA Path960 is widely used in the medical domain for the development and evaluation of image classification and disease detection algorithms, particularly in applications related to cancer diagnosis.

Dataset Organization:

Different organizational strategies were applied to the dataset to better leverage its structure and enhance the depth of the study.

3. Development environment

This section presents the development environment used to implement our image classification approach. It highlights the chosen programming language and the development platform adopted throughout the project.

3.1 Programming Language: Python

The programming language used in this work is Python, a widely adopted open-source language, particularly popular in the fields of artificial intelligence and image

processing. Python provides a rich ecosystem of powerful libraries such as NumPy, Pandas, Matplotlib, scikit-learn, and OpenCV, which facilitate data manipulation, visualization, and the implementation of machine learning algorithms.

3.2 Development platform: google colab

The development was carried out on Google Colab (Colaboratory), a free cloud-based Jupyter notebook environment provided by Google. It allows Python code to be executed online, with built-in support for popular data science libraries. Google Colab also offers seamless integration with Google Drive, making it easy to manage and access datasets and project files. In the absence of a local GPU, Colab provides a practical solution for running computations on CPU, with occasional access to cloud-based GPU/TPU resources.

3.3 Hardware Environment

The experiments were conducted on a personal computer with the following specifications:

- **Processor:** Intel(R) Core(TM) i5-2400 CPU @ 3.10 GHz ;
- **Installed RAM:** 4.00 GB ;
- **Operating System:** Windows 10, 64-bit, x64-based processor.

4. Used Libraries and tools

This section presents the main libraries and tools used during the implementation of our image classification system.

4.1 Machine Learning Libraries

- **TensorFlow**: An open-source framework developed by Google for deep learning and neural network models. It was used to build, train, and save the Convolutional Neural Network (CNN) model.
- **Keras**: A high-level API built on top of TensorFlow, providing simple interfaces for creating and training deep learning models. It was used for defining the CNN architecture and loading the saved model.
- **Scikit-learn**: Used for label encoding, performance evaluation, and general utilities such as splitting datasets.

4.2 Image Processing Libraries

- **OpenCV (cv2):** A powerful library for real-time image processing and computer vision. It was used for reading and resizing images, as well as converting image formats.
- **Pillow (PIL):** Used to display images in the GUI (for example, with Gradio or Tkinter), and to convert them into formats compatible with model input.
- **NumPy:** Used to manipulate image data as numerical arrays (matrices), especially for preprocessing steps like normalization and reshaping.

4.3 Data Manipulation Libraries

- **Pickle**: Used to save and load the label encoder (LabelEncoder object), ensuring consistency between training and inference.
- **OS**: Standard Python library used for file and path handling (checking file existence, loading paths, etc.).
- **Pandas (optional)**: In some cases, it may be used for managing datasets or organizing classification results in tabular form (e.g., CSV export).

5. Used Method

In this project, we conducted a comparative study using five classifiers: **KNN, SVM, Naïve Bayes, Decision Tree, and Random Forest**. Each classifier was applied with the following feature extraction methods:

- **LBP (Local Binary Pattern)**: A robust and efficient texture descriptor applied to the the dataset with an 80% training and 20% testing split, the dataset with an 50% training and 50% testing split, and the dataset with an 20% training and 80% testing split.
- **Haralick Features**: Used to extract texture information from images, applied to all three datasets.
- **HSV (Hue, Saturation, Value)**: A color-based method applied to Dataset, Dataset Half, and Dataset Mini.
- **Color Moments**: A statistical color descriptor used with the three datasets.
- **Fourier Descriptor**: A shape-based feature extraction method applied to Dataset, Dataset Half, and Dataset Mini.

Finally, we will adapt a CNN to our Corel1k and Kimia Path960 datasets.

We used Google's **TensorFlow** library, as well as the **Keras** library.

We will explain some descriptive details of this classifier.

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(num_classes, activation='softmax')
])
```

Figure22 Layers used for classification

- **Conv2D**: A convolutional layer that applies filters (32, then 64) to the input image. These filters help extract important features.

- **MaxPooling2D:** A pooling layer that reduces the spatial dimensions of the feature maps by taking the maximum value from small regions (usually 2x2). This helps reduce computation and retain the most relevant features.

- **Flatten**: This layer flattens the 2D feature maps into a 1D vector. It prepares the data for the fully connected layers.

- **Dense**: Fully connected layers inspired by biological neurons.

- **Dropout (0.5)**: A regularization technique that randomly deactivated 50% of the neurons during training. This helps prevent overfitting by ensuring the model doesn't rely too heavily on any one neuron.

- **activation='relu'**: The ReLU (Rectified Linear Unit) activation function introduces non-linearity into the model. It replaces all negative values with zero, allowing the network to learn more complex patterns.

- **activation='softmax'**: The Softmax activation function is used in the output layer to convert raw prediction scores into probabilities for each class in a multi-class classification problem.

6. Results and Interpretation

6.1 Results of experiments using the Corel 1K database:

In this chapter we present graphs that summarize the results obtained:

Graph of results using the kNN classifier:

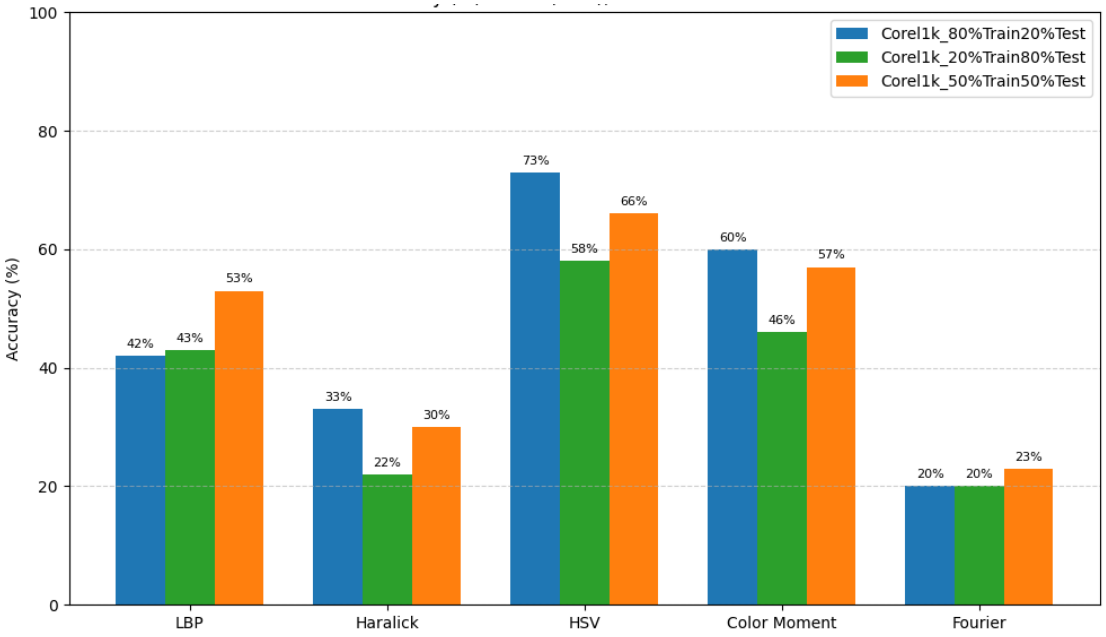

Figure23 Graph showing the results of the KNN classifier

We note here that the best method used is HSV whatever the base used. An **accuracy of up to 73%** was achieved when using the original dataset with a **20% test split**.

Thus, knowledge was extracted from these experiments.

Knowledge 1: In image mining, using the K-Nearest Neighbors (KNN) classifier with Euclidean distance, this study has shown that the HSV color space often provides the best results when compared to other color spaces.

Class-Wise Results The results of the best classification method used so far are detailed: the KNN Classifier with Color Histogram, Euclidean Distance, and the complete HSV color space. This detail is illustrated graphically in Figure 24.

The graph shows the percentage of correct predictions for each class, depending on the dataset used.

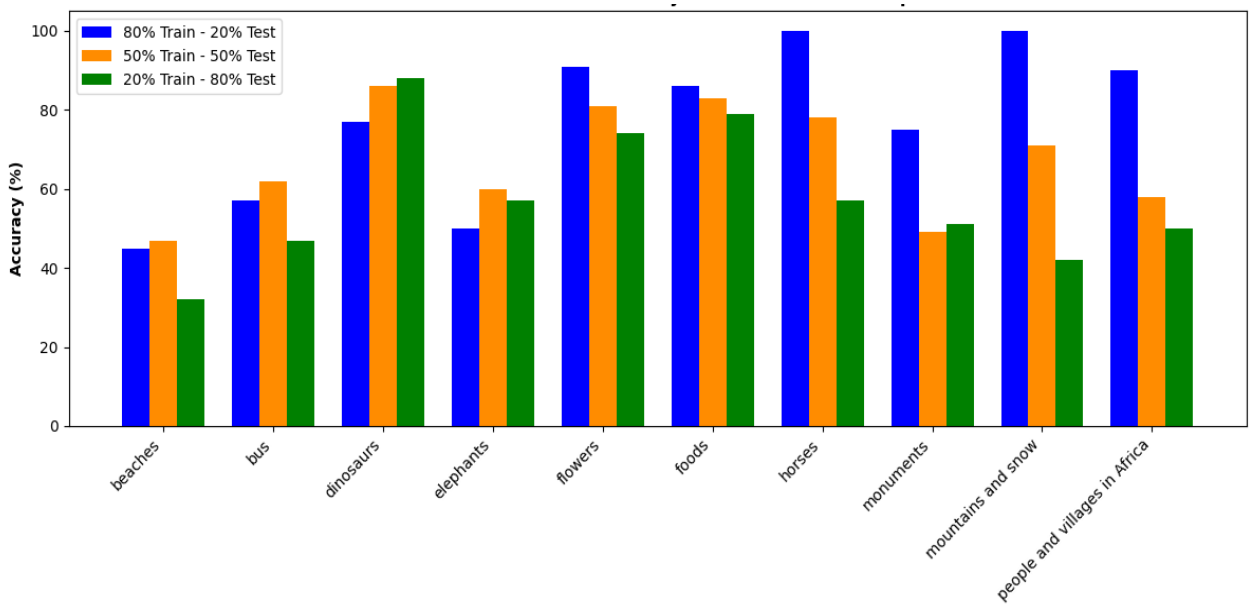

Figure24 Class-wise results of the best-performing classifier so far: KNN –Euclidean – Full HSV space

We then notice that, when using the original database, we obtain in some cases such as 'horses' and 'mountains and snow' a percentage of 100%. All of this can be interpreted as a second piece of knowledge extracted in this study.

Knowledge 2: Regarding the Corel1000 dataset, using the KNN classifier with Euclidean distance and HSV color space, some classes are easier to classify than others. Generally speaking, the easiest classes to classify are those that are not overly rich in concepts.

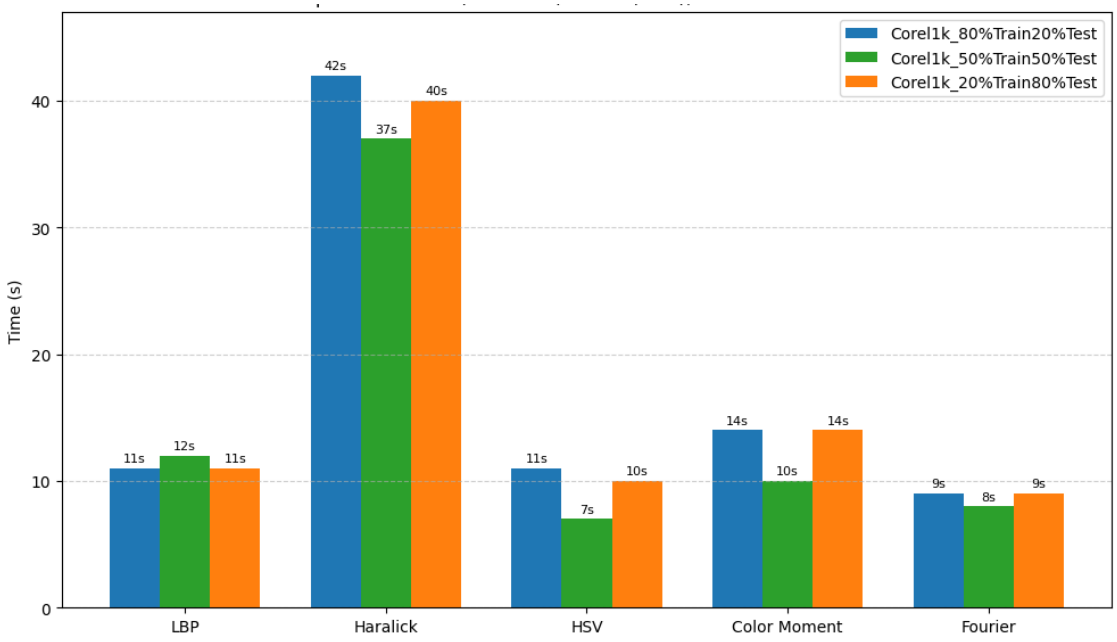

Figure25 Graph showing the execution time of the KNN classifier

Graph of results using the SVM classifier:

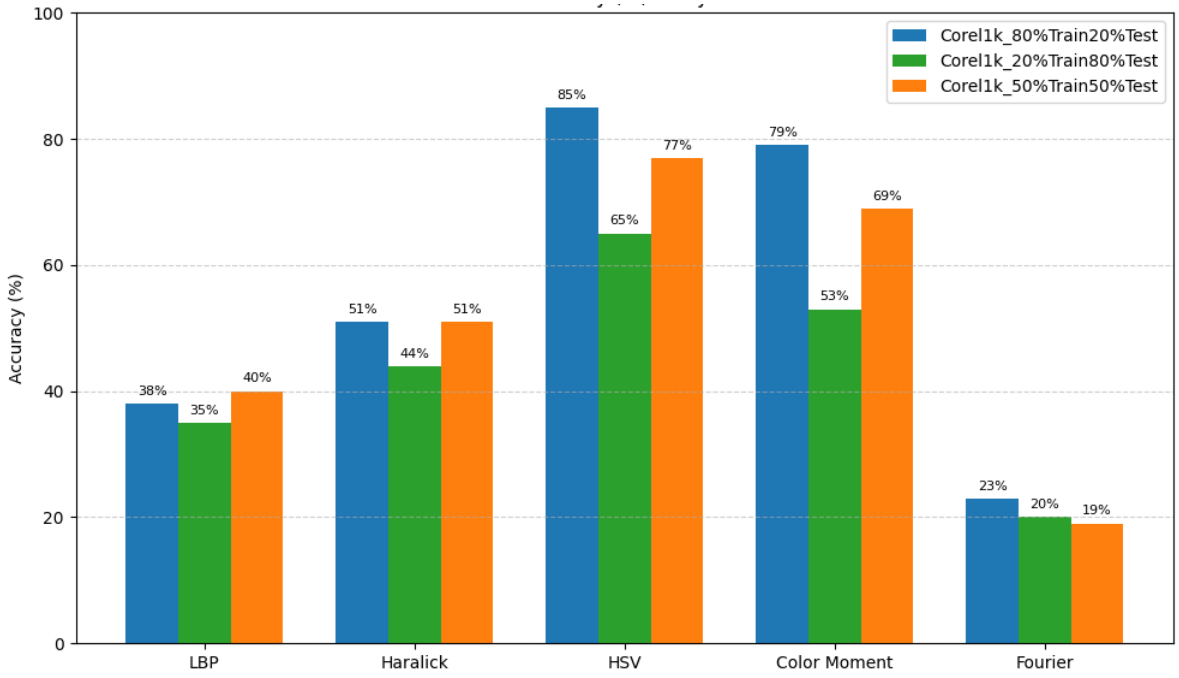

Figure26 Graph showing the results of the SVM classifier

We note that the best-performing method is HSV, regardless of the dataset used. An accuracy of up to 85% was achieved when using the original dataset with a 20% test split. Therefore, this outcome can be considered as knowledge extracted from the experiments.

Knowledge 3: In image mining, using the Support Vector Machine (SVM) classifier, the HSV color space has proven to be the most effective for classification. The best results were achieved with the linear kernel and a regularization parameter C=1.0. In this configuration, the accuracy obtained on the Corel1K dataset with a 20% test split reached up to 85%, the accuracy remained high, further highlighting the effectiveness of the HSV color space in image mining.

Detailed results by class (Class-Wise): We will detail the results of the best classification method used so far: the SVM Classifier – HSV Space. This detail is illustrated graphically in Figure 27.

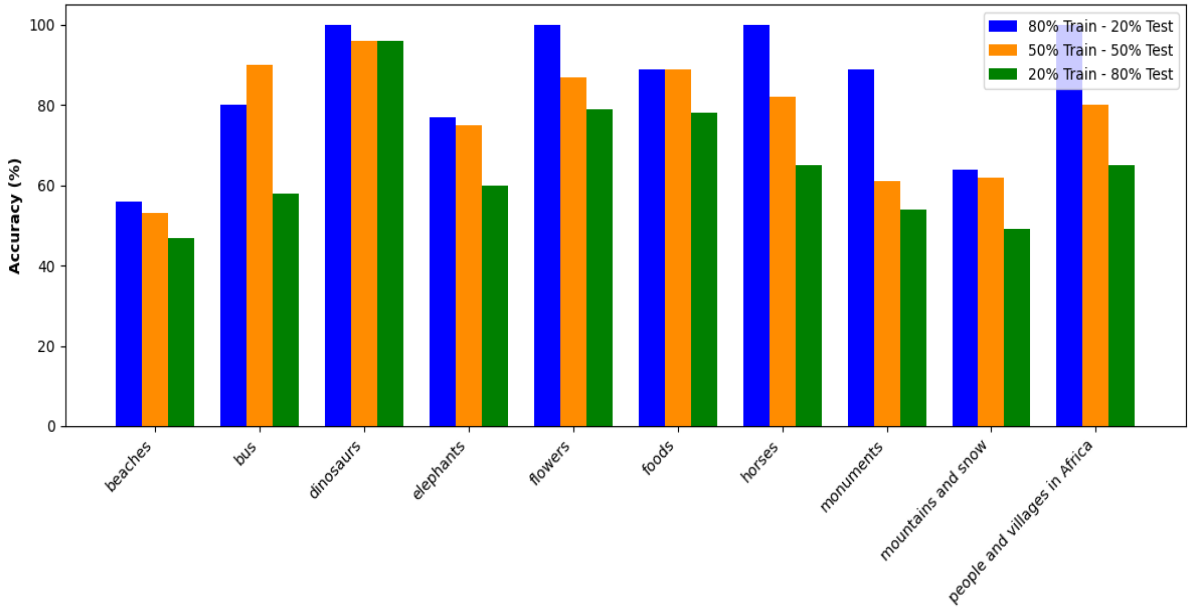

Figure27 Class-wise results of the best-performing classifier so far: SVM– Full HSV space

We then notice that when we use the initial database, we get in some cases, such as "dinosaurs","flowers"," horses" and "people_and_villages_in_Africa a percentage of 100% can be interpreted as second knowledge, extracted in this study.

Knowledge 4: Regarding the Corel1000 Dataset, using the SVM classifier (HSV color space), some classes are easier to classify than others. Generally speaking, the easiest classes to classify are those that are not overly rich in concepts.

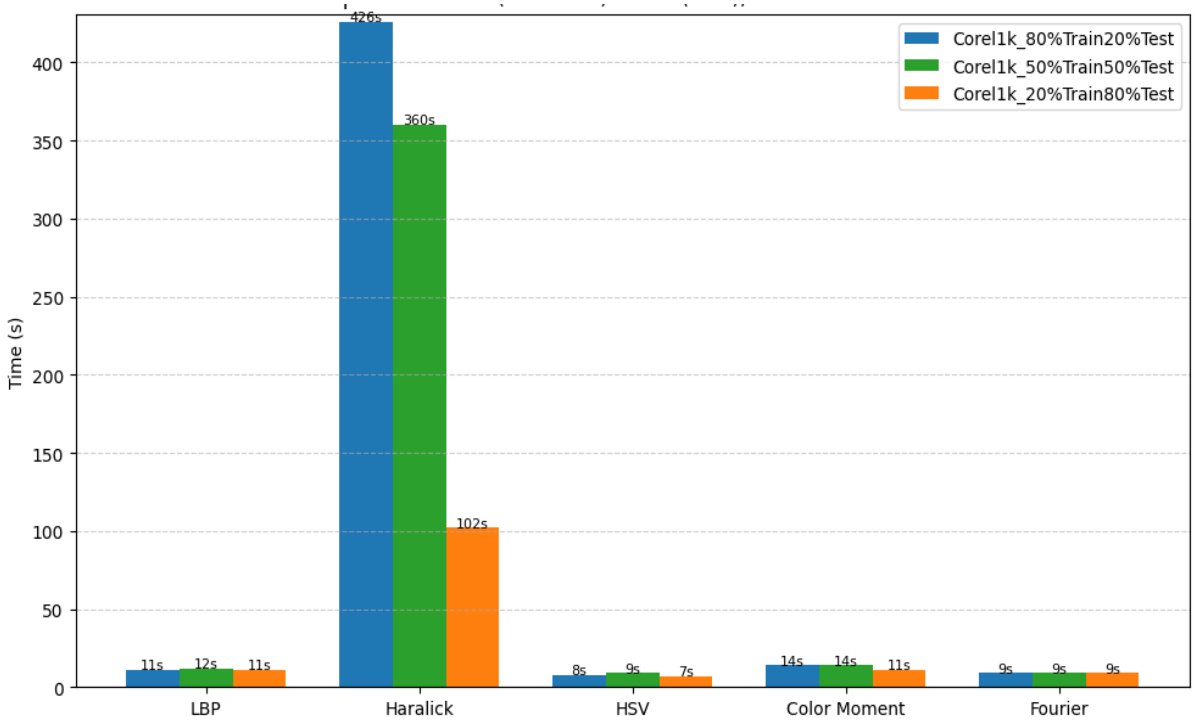

Figure28 Graph showing the execution time of the SVM classifier

Graph of results using the Bayes Naïve classifier:

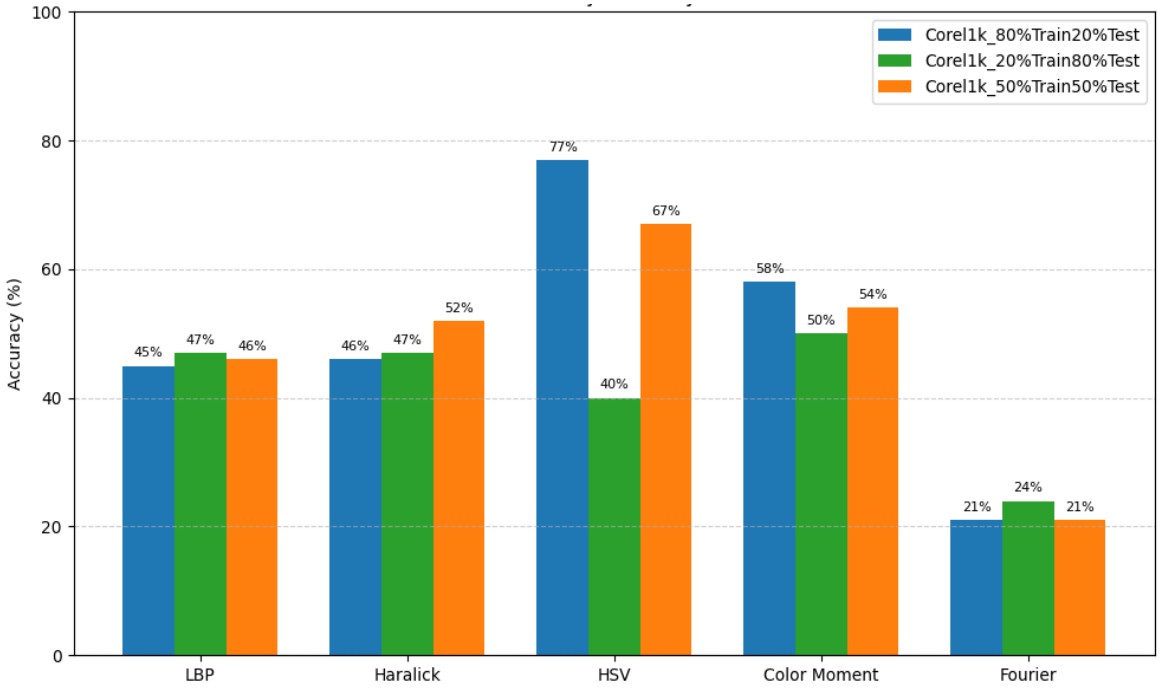

Figure29 Graph showing the results of the Bayes Naive classifier

We note that the best-performing method is HSV, regardless of the dataset used. An accuracy of up to 77% was achieved when using the original dataset with a 20% test split. Therefore, this outcome can be considered as knowledge extracted from the experiments.

Knowledge 5: In image mining, using the Bayes Naïve classifier, the best color space is HSV.

Detailed results by class (Class-Wise): We will detail the results of the best classification method used so far: the Bayes Naïve classifier – HSV Space. This detail is illustrated graphically in Figure 30.

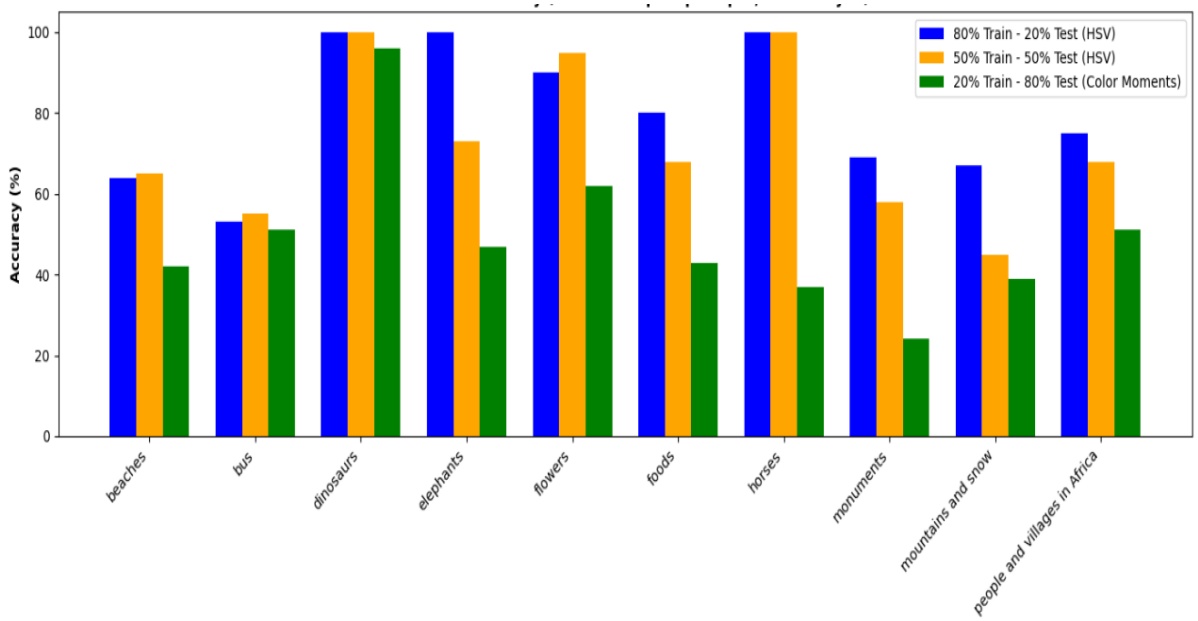

Figure30 Class-wise results of the best-performing classifier so far: Bayes Naive– Full HSV space

We then notice that when we use the initial database, we get in some cases, such as "dinosaurs","elephants, and" horses" a percentage of 100% can be interpreted as second knowledge, extracted in this study.

Knowledge 6: Regarding the Corel1000 Dataset, using the Bayes Naive classifier (HSV color space), some classes are easier to classify than others. Generally speaking, the easiest classes to classify are those that are not overly rich in concepts.

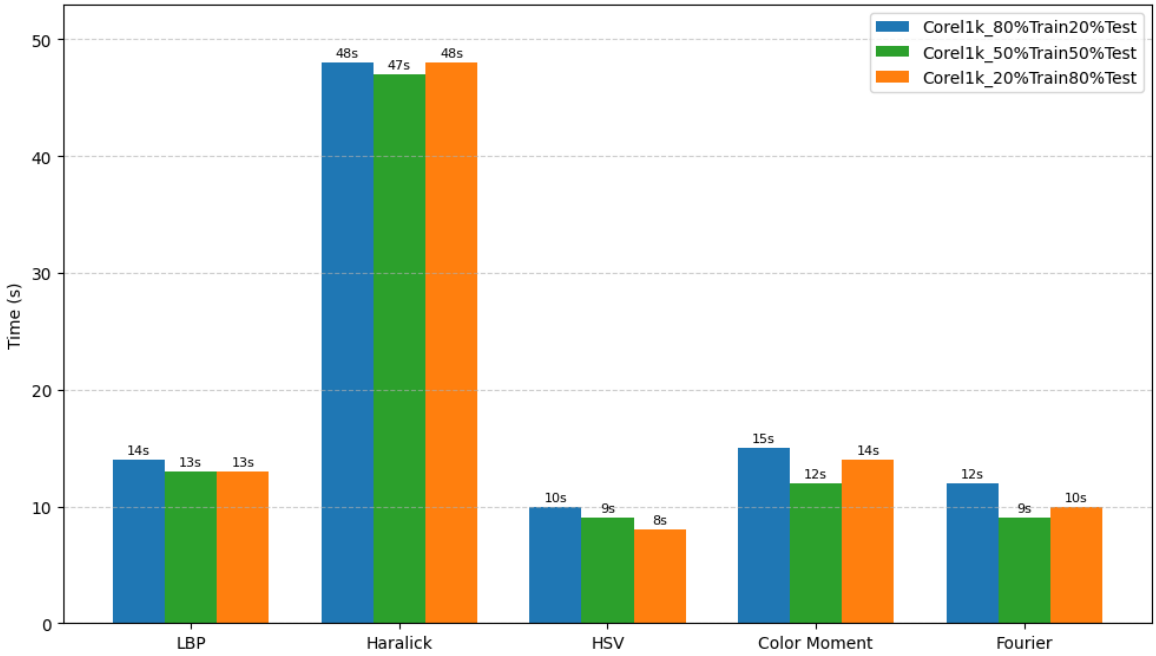

Figure31 Graph showing the execution time of the Bayes Naïve classifier

Graph of results using the Decision Tree classifier:

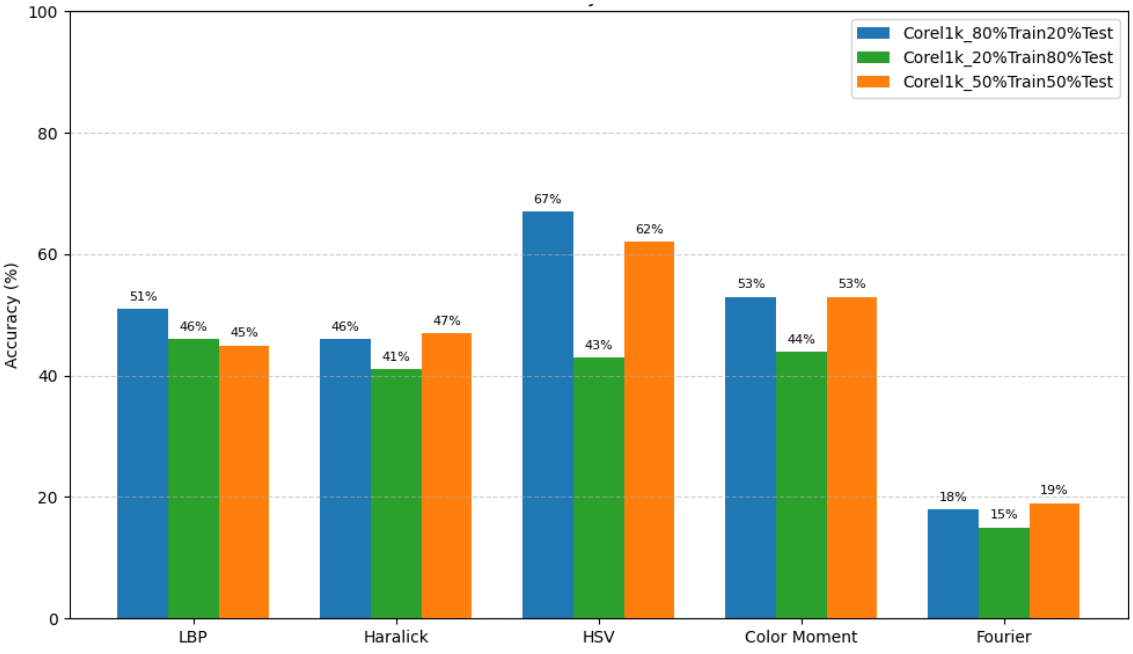

Figure32 Graph showing the results of the Decision Tree classifier

We observed that the best-performing descriptor for the **Decision Tree** classifier is the **HSV color descriptor**, regardless of the dataset split. An accuracy of **67%** was achieved using

the original dataset with a **20% test split**, while the accuracy dropped to **62%** with a **50% test split**, and further to **46%** when using an **80% test split** with the **LBP descriptor**.

Knowledge 7: In image mining, using the Decision tree classifier, the best color space is HSV.

Detailed results by class (Class-Wise): We will detail the results of the best classification method used so far: the Decision tree Classifier – HSV Space. This detail is illustrated graphically in Figure 33.

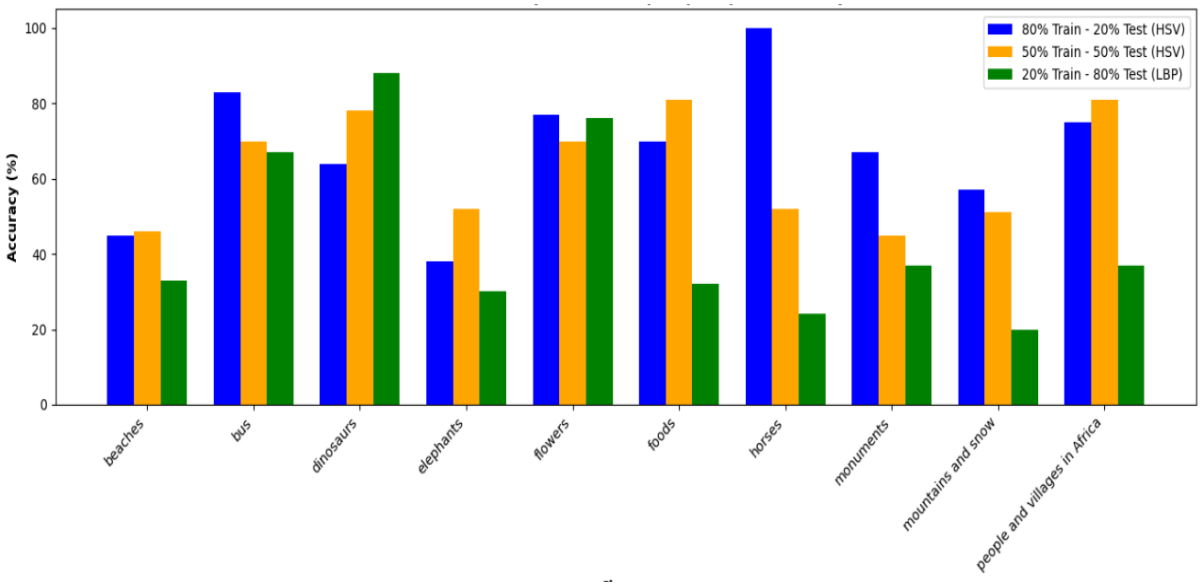

Figure33 Class-wise results of the best-performing classifier so far: Decision Tree– Full HSV space

We then notice that when we use the initial database, we get in some cases, such as" horses" a percentage of 100% can be interpreted as second knowledge, extracted in this study.

Knowledge 8: Regarding the Corel1000 Dataset, using the Decision tree classifier (HSV color space), some classes are easier to classify than others. Generally speaking, the easiest classes to classify are those that are not overly rich in concepts.

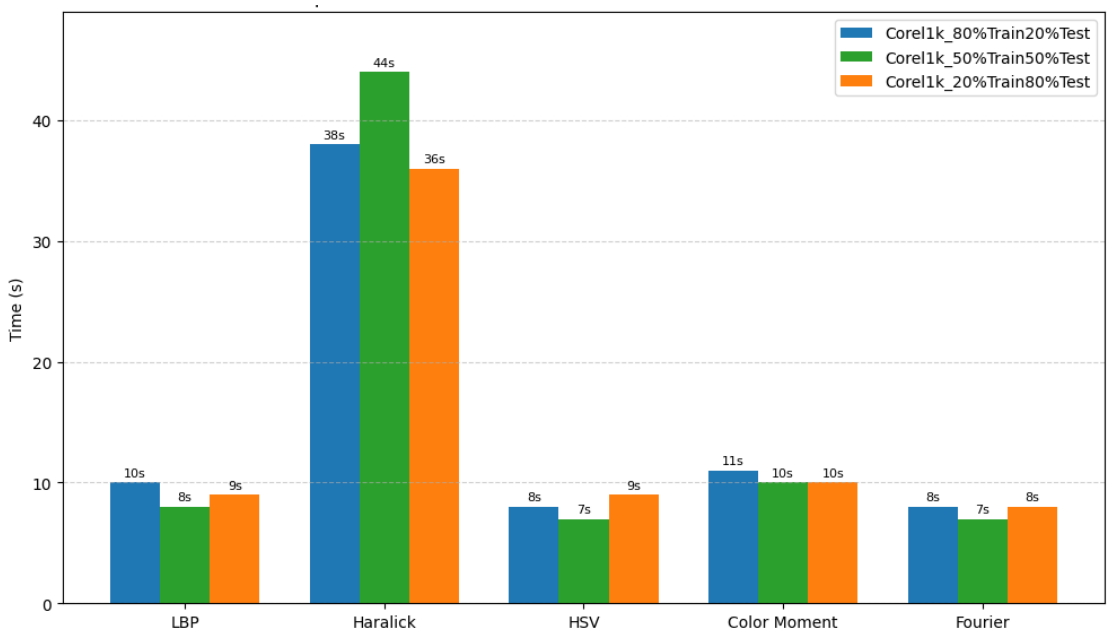

Figure34 Graph showing the execution time of the Decision Tree classifier

Graph of results using the Random Forest classifier:

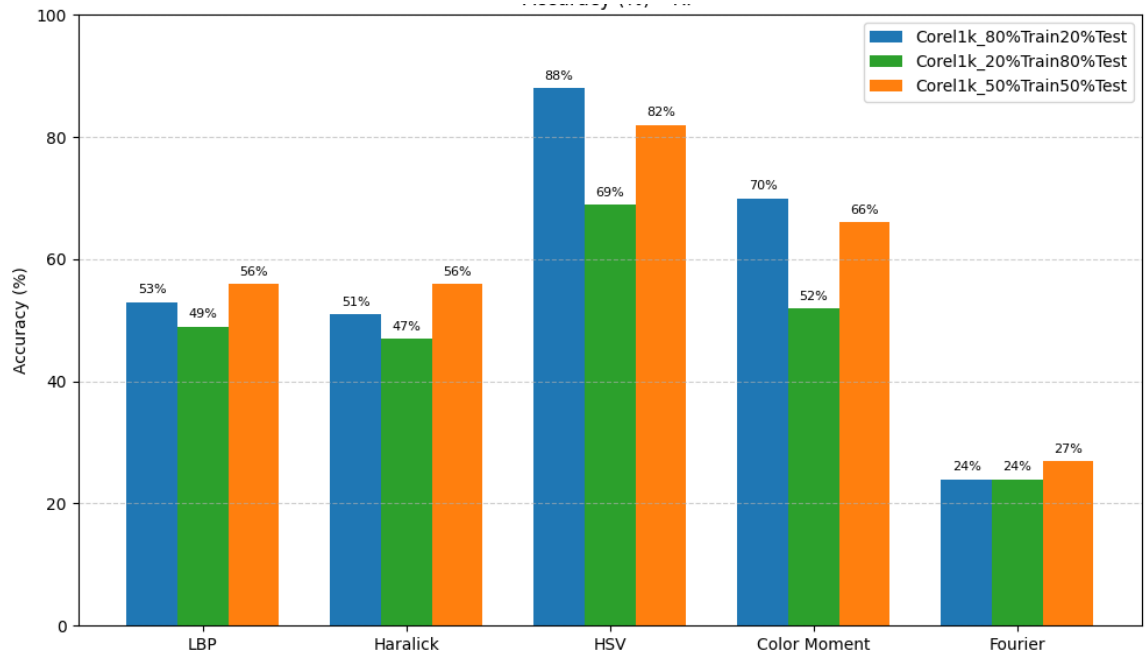

Figure35 Graph showing the results of the Random Forest classifier

We observed that the best-performing descriptor for the **Random Forest** classifier is the **HSV color descriptor**, regardless of the dataset split. An accuracy of **88%** was achieved

using the original dataset with a **20% test split**, while the accuracy dropped to **82%** with a **50% test split**, and further to **69%** when using an **80% test split**.

Knowledge 9: In image mining, using the Random Forest classifier, the best color space is HSV.

Detailed results by class (Class-Wise): We will detail the results of the best classification method used so far: the Random Forest Classifier – HSV Space. This detail is illustrated graphically in Figure 36.

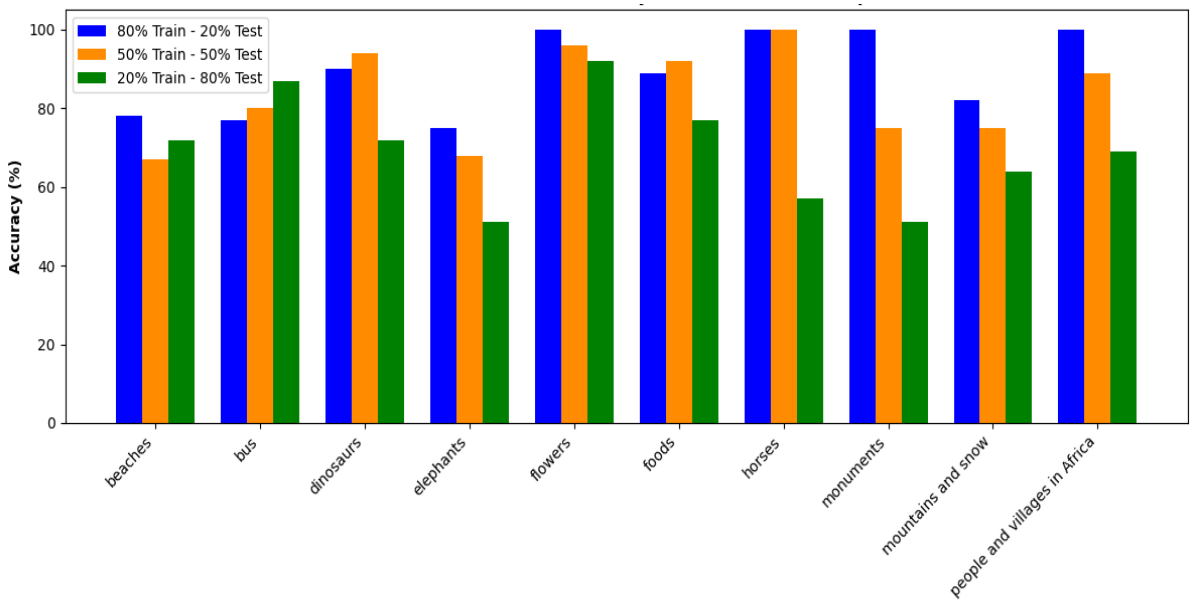

Figure36 Class-wise results of the best-performing classifier so far: Random Forest– Full HSV space

We then notice that when we use the initial database, we get in some cases, such as “flowers”," horses", " monuments” and “people_and_villages_in_Africa” a percentage of 100% can be interpreted as second knowledge, extracted in this study.

Knowledge 10: Regarding the Corel1000 Dataset, using the Random Forest classifier (HSV color space), some classes are easier to classify than others. Generally speaking, the easiest classes to classify are those that are not overly rich in concepts.

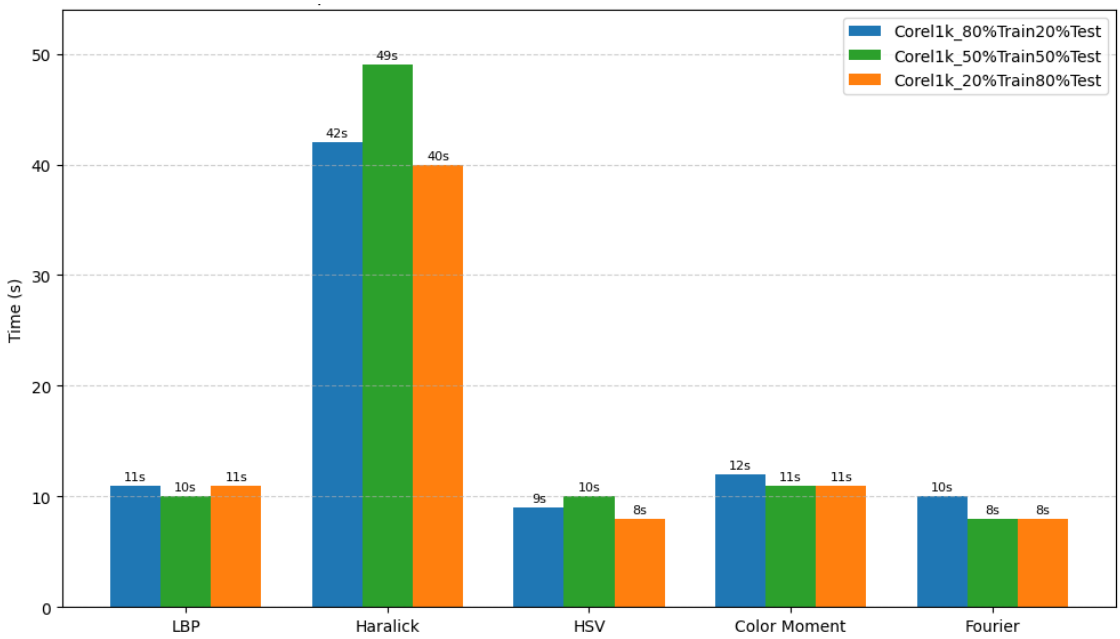

Figure37 Graph showing the execution time of the Random Forest classifier

Knowledge 11: In image mining, experiments conducted on the Corel-1000 dataset using five classifiers K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naïve Bayes, Decision Tree, and Random Forest with handcrafted features based on the Fourier Transform have shown very low accuracy, ranging from 15% to 27%.

This confirms that, generally speaking, classifiers based on Fourier descriptors are weak classifiers, especially when dealing with images rich in concepts.

The reason lies in the nature of Fourier descriptors, which primarily capture global shape information, but are less sensitive to local details, textures, and colors.

Graph of results using CNN:

So, we applied CNN to our Corel 1K database. We performed 10 epochs on the training data set.

The training results are graphically represented in Figure.

This figure illustrates the progression of the correct classification rate (Accuracy) as a function of the number of epochs.

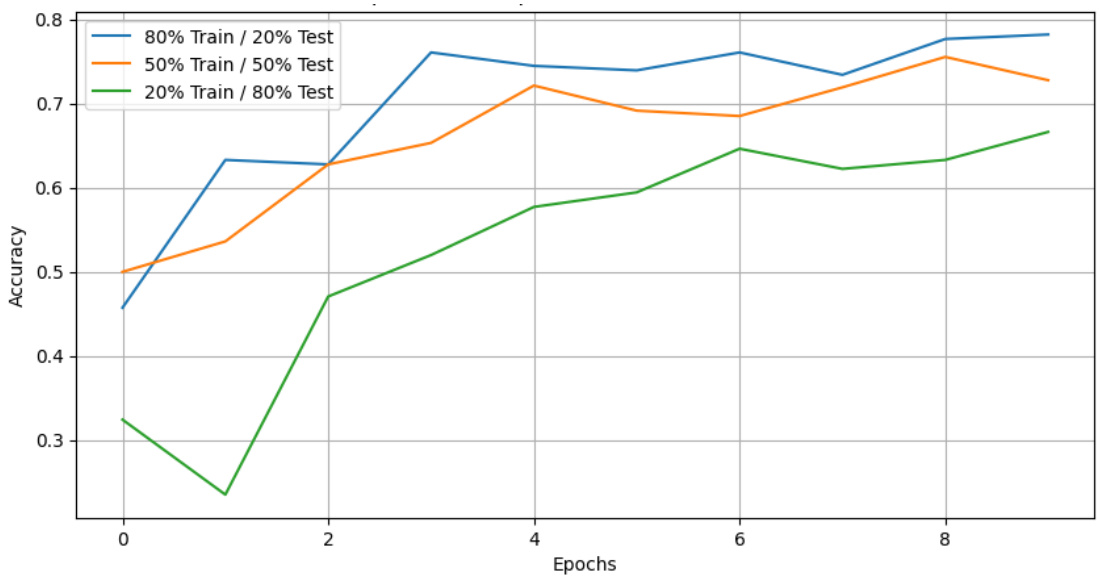

Figure38. Evolution of CNN classification accuracy as a function of the number of epochs

We observe an improvement in accuracy over the epochs, reaching up to 78.19% when using 80% training samples. However, as the training set becomes smaller, the accuracy decreases from 78.19% to 72.77% with 50% training samples, and 66.62% with 20% samples. This leads us to another insight gained from this study.

Knowledge 12: In the context of image mining, the CNN-based classifier implemented with TensorFlow is sensitive to the number of training epochs and the amount of training data used.

Knowledge 13: In the context of the Corel-1K dataset, which is rich in concepts, Convolutional Neural Networks (CNNs) generally demonstrate higher classification efficiency, except for handcrafted feature-based methods using the HSV color space When combined with classifiers Random Forest and Support Vector Machines (SVM).

6.2 Results of experiments using the Kimia path 960 database:

In this section we present graphs that summarize the results obtained:

Graph of results using the kNN classifier:

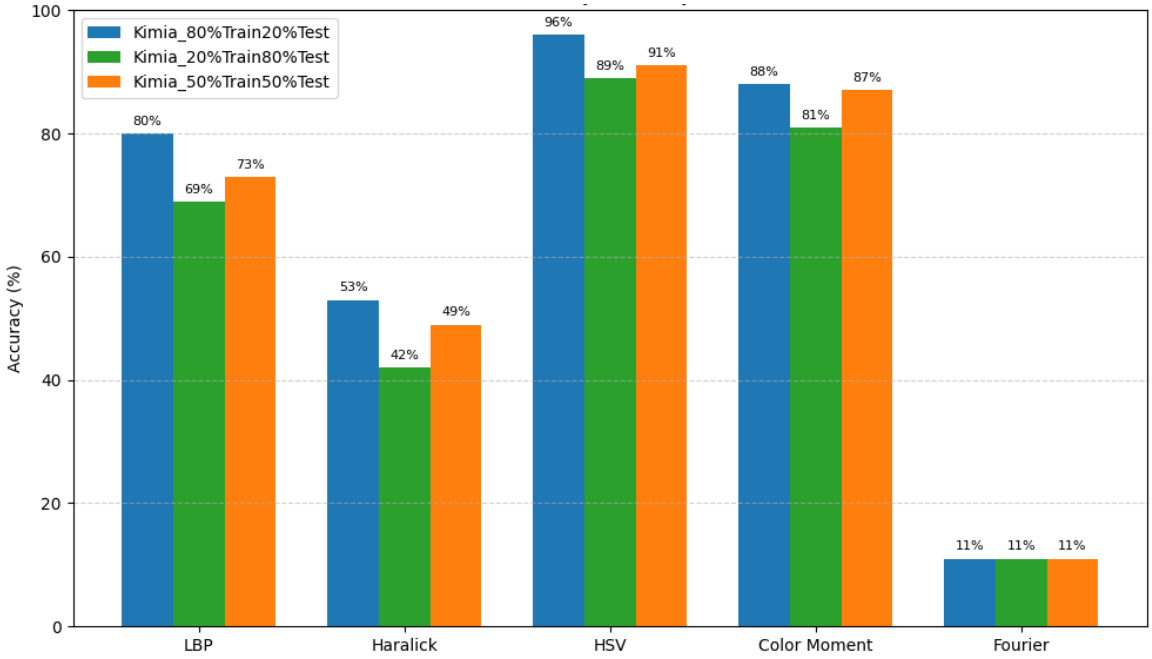

Figure39 Graph showing the results of the KNN classifier

We note here that the best method used is HSV whatever the base used.

Knowledge 14: we observe that the best method used is HSV with accuracy closely approaching 100%, which combines all the axes. We achieve up to 96% accuracy for the kimia-80% train 20% test dataset, up to 91% accuracy for the kimia-50% train 50% test dataset, and up to 89% accuracy for the kimia-20% train 80% test dataset.

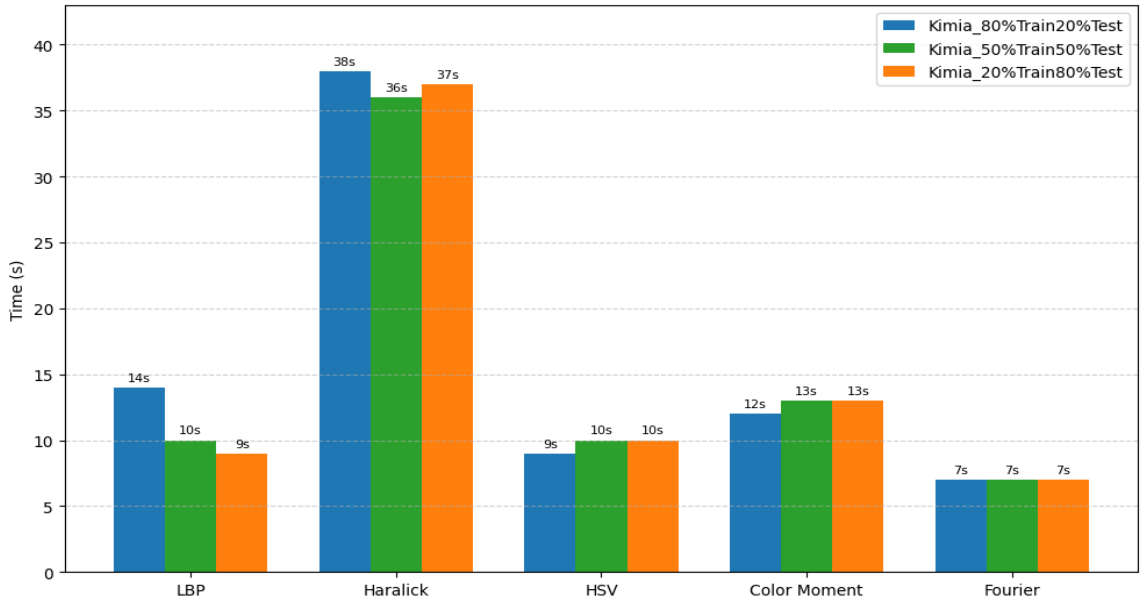

Figure40 Graph showing the execution time of the KNN classifier

Graph of results using the SVM classifier:

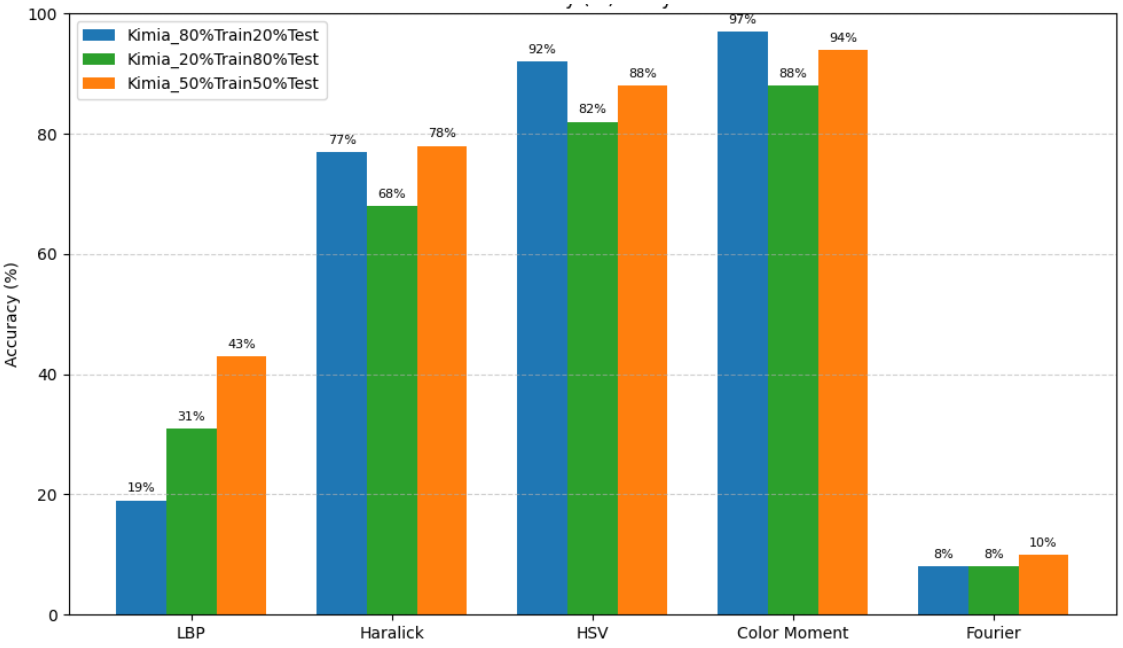

Figure41 Graph showing the results of the SVM classifier

We note here that the best method used is Color Moment whatever the base used.

Knowledge 15: we observe that the best method used is Color Moment. We achieve up to 97% accuracy for the kimia-80%Train20% Test dataset, up to 94% accuracy for the kimia-50%Train 50%Test dataset, and up to 88% accuracy for the kimia-20%Train 80%Test dataset.

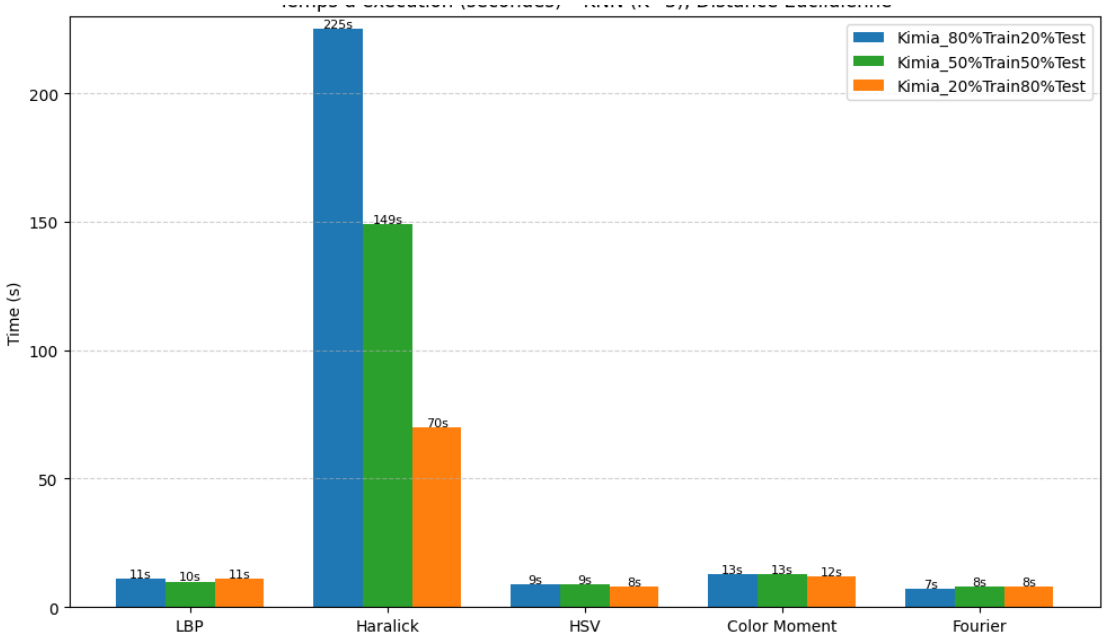

Figure42 Graph showing the execution time of the SVM classifier

Graph of results using the Bayes Naïve classifier:

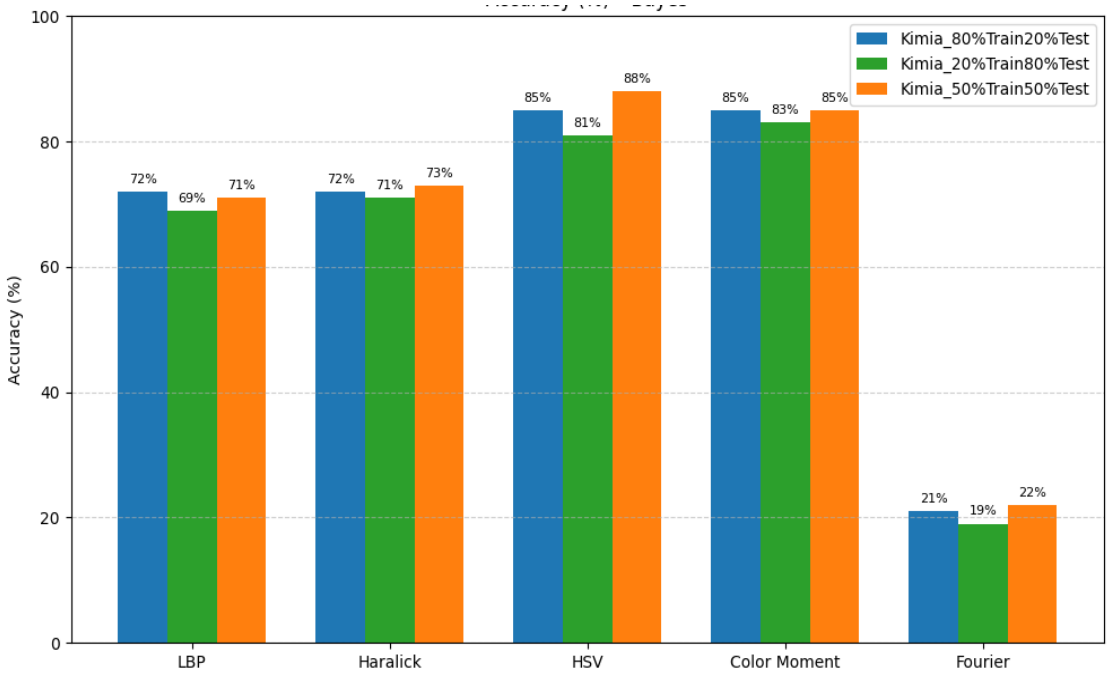

Figure43 Graph showing the results of the Bayes Naive classifier

Here too we notice that the best method used is HSV.

Knowledge 16: When using the Kimia dataset with an 80% training and 20% testing split, we observe that the best-performing methods are HSV and Color Moments, each achieving up to 85% accuracy.

Knowledge 17: When using the Kimia dataset with a 50% training and 50% testing split, we observe that the best-performing method is HSV, achieving 88% accuracy, followed by Color Moments with 85% accuracy.

Knowledge 18: When using the Kimia dataset with a 20% training and 80% testing split, we observe that the best-performing method is Color Moment, achieving 83% accuracy, followed by HSV with 81% accuracy.

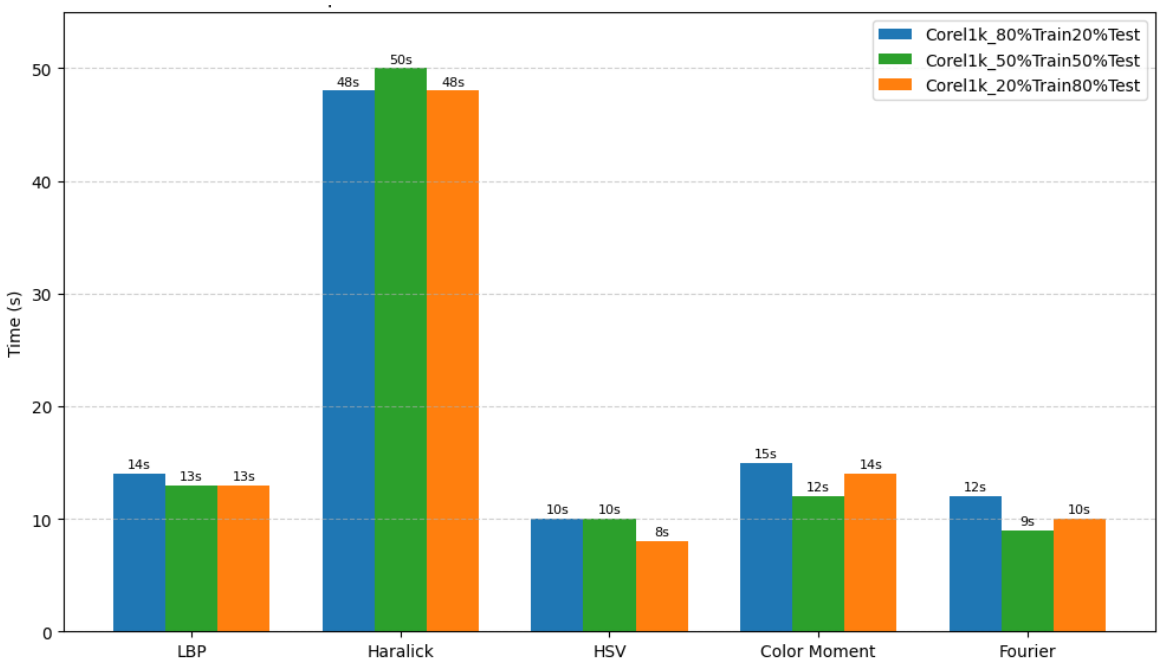

Figure44 Graph showing the execution time of the Bayes Naive classifier.

Graph of results using the Decision Tree classifier:

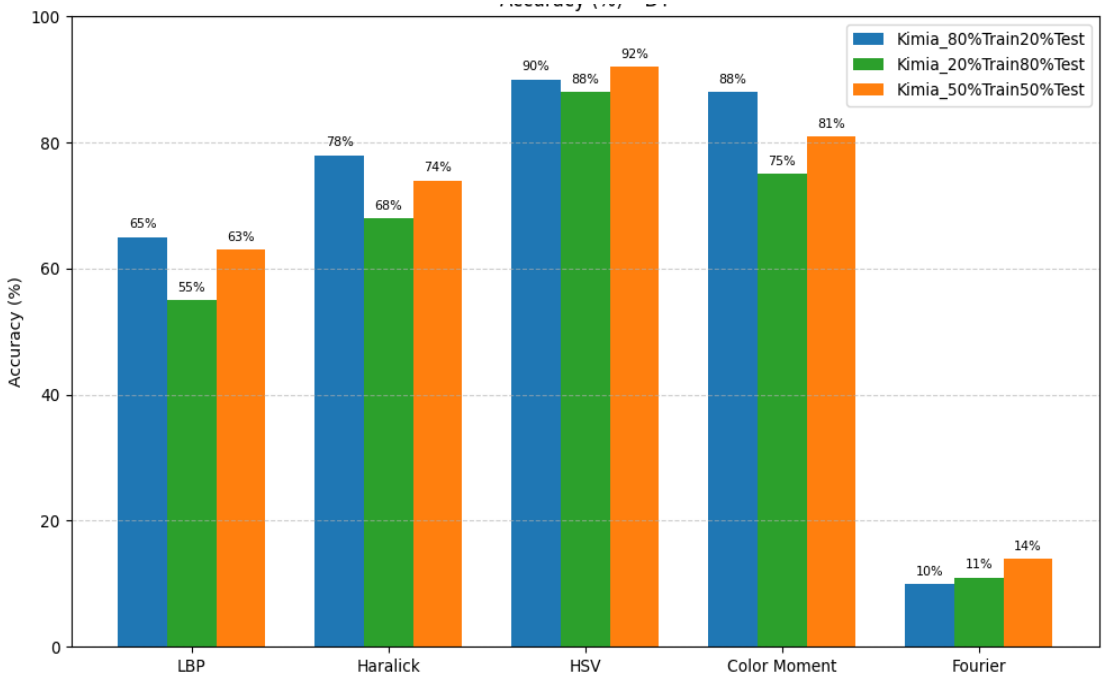

Figure45 Graph showing the results of the Decision Tree classifier

Here too we notice that the best method used is HSV.

Knowledge 19: When using the Kimia dataset with an 80% training and 20% testing split, we observe that the best-performing method is HSV, achieving 90% accuracy, followed by Color Moments with 88% accuracy.

Knowledge 20: When using the Kimia dataset with a 50% training and 50% testing split, we observe that the best-performing method is HSV, achieving 92% accuracy, followed by Color Moments with 81% accuracy.

Knowledge 21: When using the Kimia dataset with a 20% training and 80% testing split, we observe that the best-performing method is HSV with 88% accuracy.

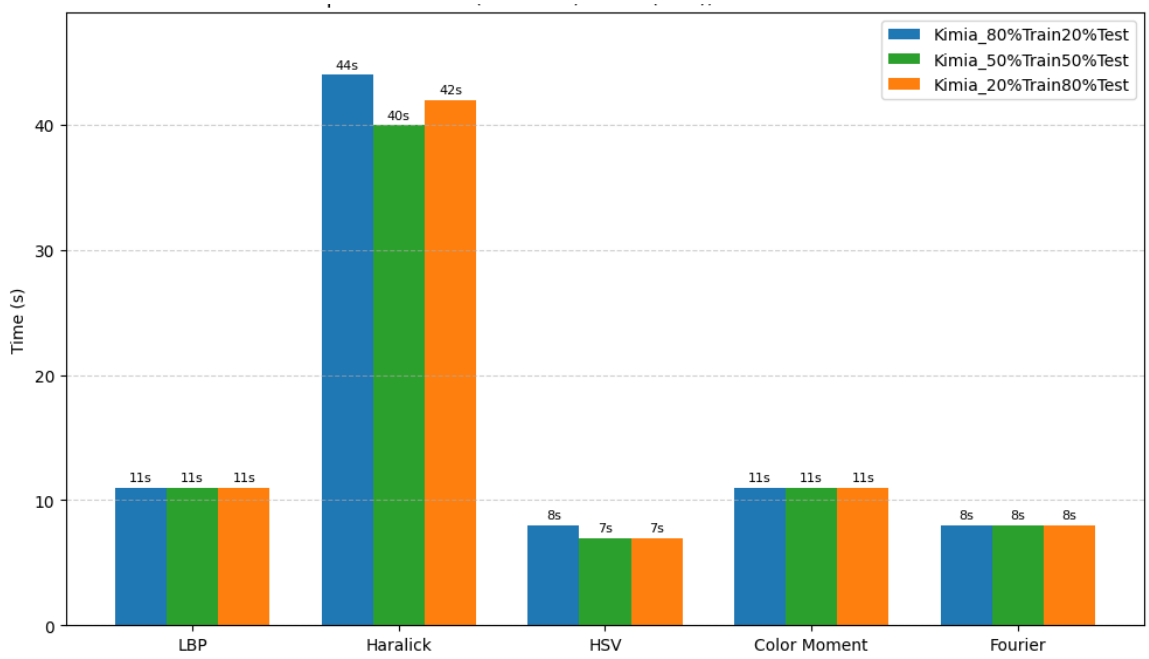

Figure46 Graph showing the execution time of the Decision Tree classifier.

Graph of results using the Random Forest classifier:

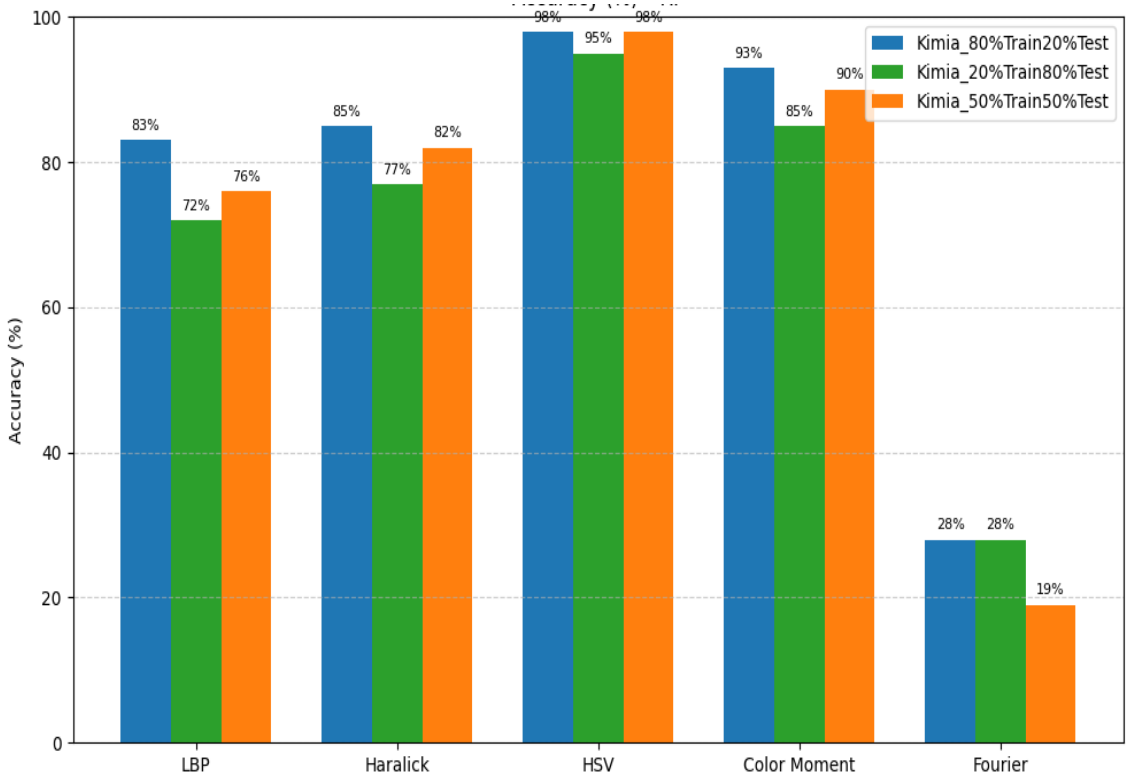

Figure47 Graph showing the results of the Random Forest classifier

Here too we notice that the best method used is HSV.

Knowledge 22: When using the Kimia dataset with an 80% training and 20% testing split, we observe that the best-performing method is HSV, which combines all the axes. It achieves up to 98% accuracy, followed by the Color Moment with 93%.

Knowledge 23: When using the Kimia dataset with an 50% training and 50% testing split, we observe that the best-performing method is HSV, which combines all the axes. It achieves up to 98% accuracy, followed by the Color Moment with 90%.

Knowledge 24: When using the Kimia dataset with an 20% training and 80% testing split, we observe that the best-performing method is HSV, which combines all the axes. It achieves up to 95% accuracy.

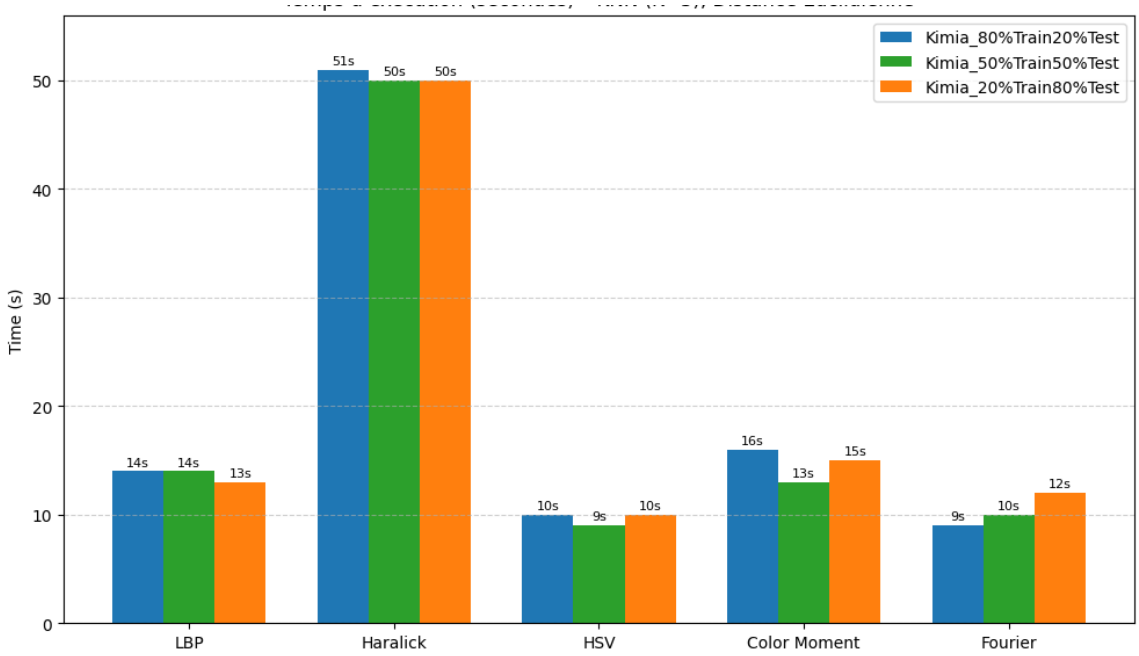

Figure48 Graph showing the execution time of the Random Forest classifier.

Knowledge 25: In the context of image mining, hand-crafted feature extraction methods tend to be more effective for images that are poor in concepts, such as texture images. Generally speaking, when the image is lacks concepts diversity, these methods can still capture enough discriminative information for classification tasks.

Knowledge 26: According to the results, using only 20% of the data for training yields average performance with LBP, but performs better especially with HSV, Haralik and Color Moments.

This suggests that handcrafted features remain robust even with limited training data.

Graph of results using CNN:

So, we applied the CNN to our Kimia path 960 database. We performed 10 epochs on the training data set.

The training results are graphically represented in Figure.

This figure illustrates the progression of the correct classification rate (Accuracy) as a function of the number of iterations.

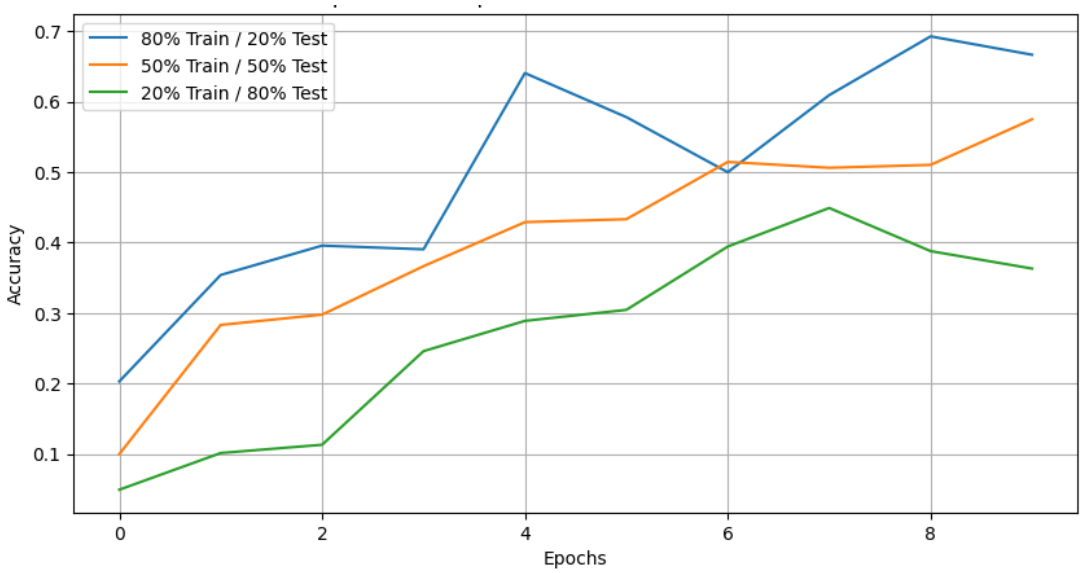

Figure49. Evolution of CNN classification accuracy as a function of the number of epochs

Knowledge 27: We observe an improvement in accuracy over the iterations, reaching up to 66.67% when using 80% training samples. However, as the training set becomes smaller, the accuracy decreases from 66.67% to 57.50% with 50% training samples, and 36.33% with 20% samples.

Knowledge 28: In the context of image mining, deep learning methods such as Convolutional Neural Networks (CNNs) tend to be less effective for images that are poor in concepts, such as texture images. Generally speaking, when the image is lacks concepts diversity, CNNs may struggle to learn meaningful representations, as they are designed to capture complex data.

Knowledge 29: By using only 20% of the data for training, the performance of some handcrafted feature-based methods (HSV and Color Moments), combined with classifiers (KNN, SVM, Naive Bayes, Decision Tree, and Random Forest) was observed to outperform CNNs, despite the latter's well-known generalization capabilities. This highlights that, with limited training data, handcrafted features can be more effective.

Meta-knowledge

After extracting knowledge from image data, an important meta-knowledge emerges, which relates to the distinction between images rich in concepts and those poor in concepts, as well as the effectiveness of deep learning methods versus handcrafted features. Generally, handcrafted feature-based approaches tend to be more effective for images that are poor in concepts. In contrast, Convolutional Neural Networks (CNNs) demonstrate significantly

higher efficiency when dealing with images that are rich in concepts. However, there are notable exceptions: color moment and HSV-based classifiers, can yield highly competitive results, even outperforming CNNs.

7. User Interface for Image Classification Using CNN Trained on Corel 1k

This section presents the graphical interface developed to test image classification using a **Convolutional Neural Network (CNN)** model trained specifically on the **Corel1k dataset**. The interface allows the user to select an image, run classification, and view the predicted class.

General View of the Interface

This part shows the main graphical interface in its default state. It includes buttons for loading and classifying an image, and a display area for the result.

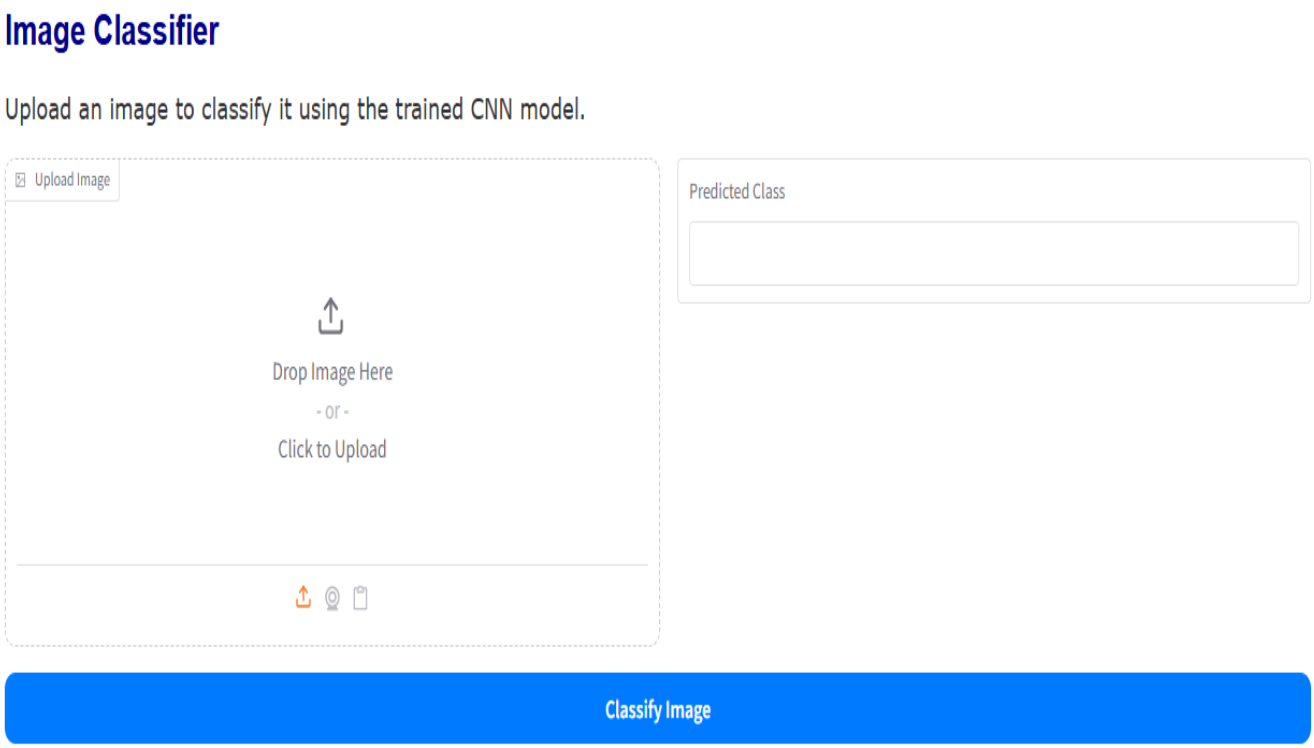

Figure50 Overview of the interface

Loading an Image

By clicking on the **"Drop Image Here"** button, the user can open a file explorer to choose an image from their local machine.

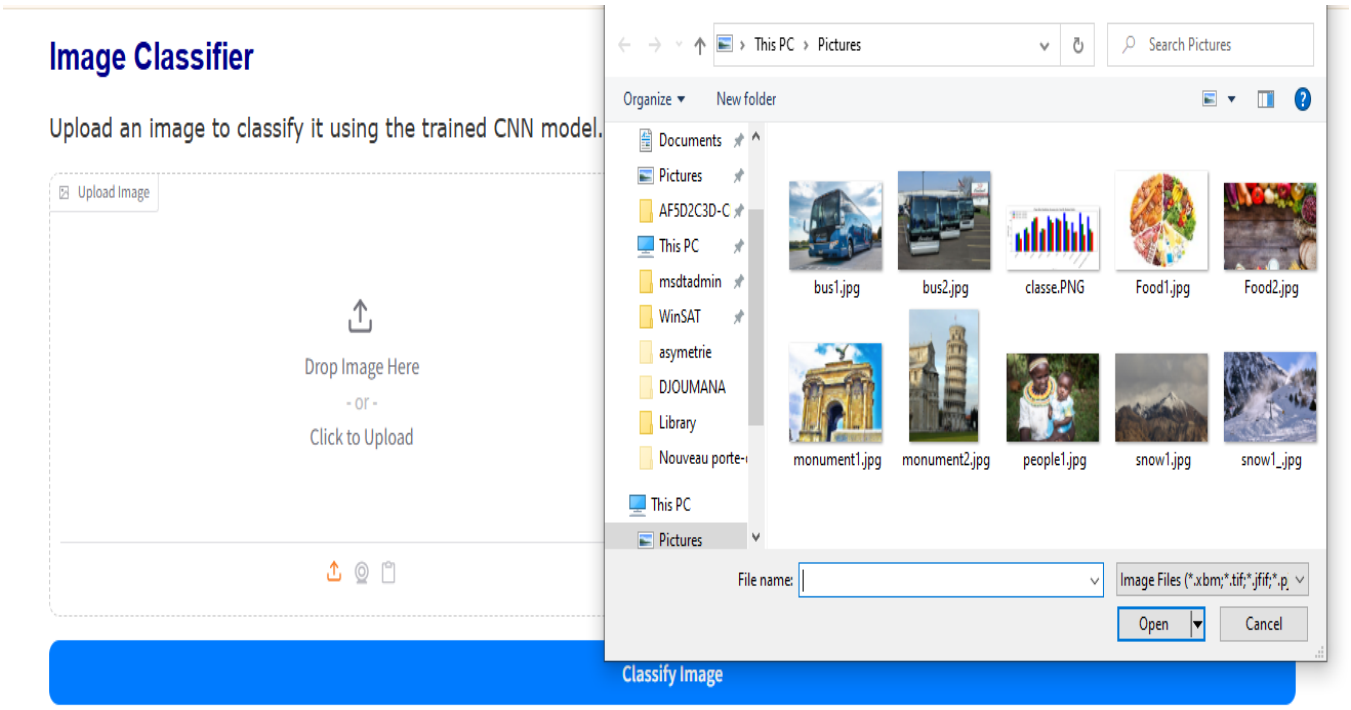

Figure51 Loading an image

The selected image is then displayed in the interface.

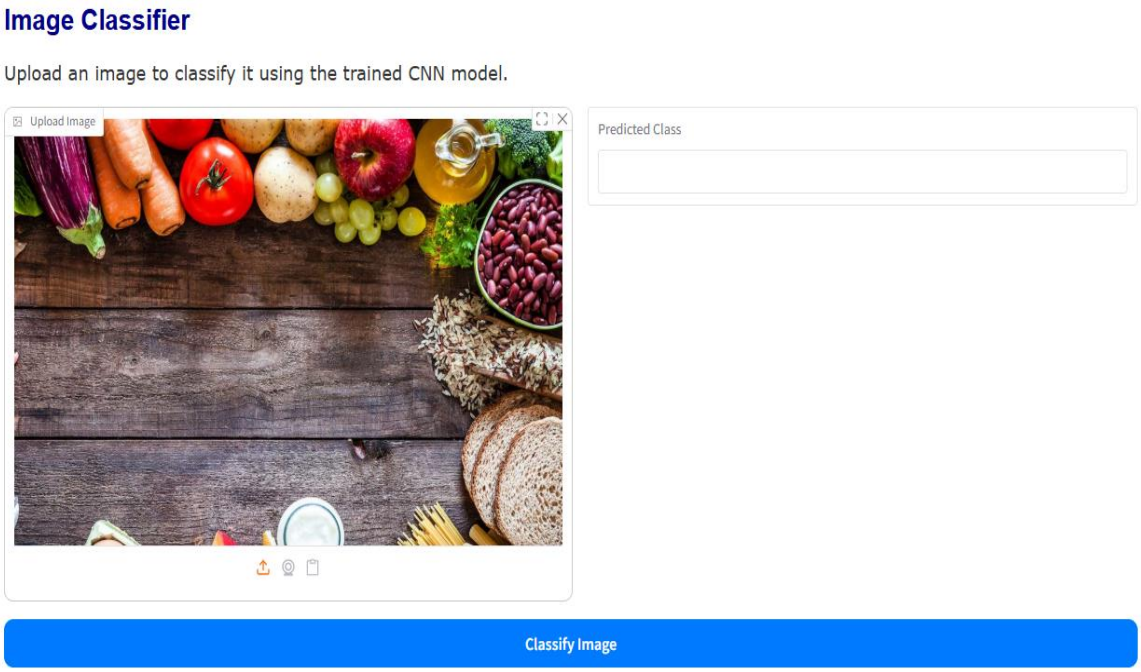

Figure52 Selected image

Classifying the Image

After loading, the user clicks on the "**Classify Image**" button. The image is passed to the CNN model trained on Corel 1k, which predicts the corresponding class label.

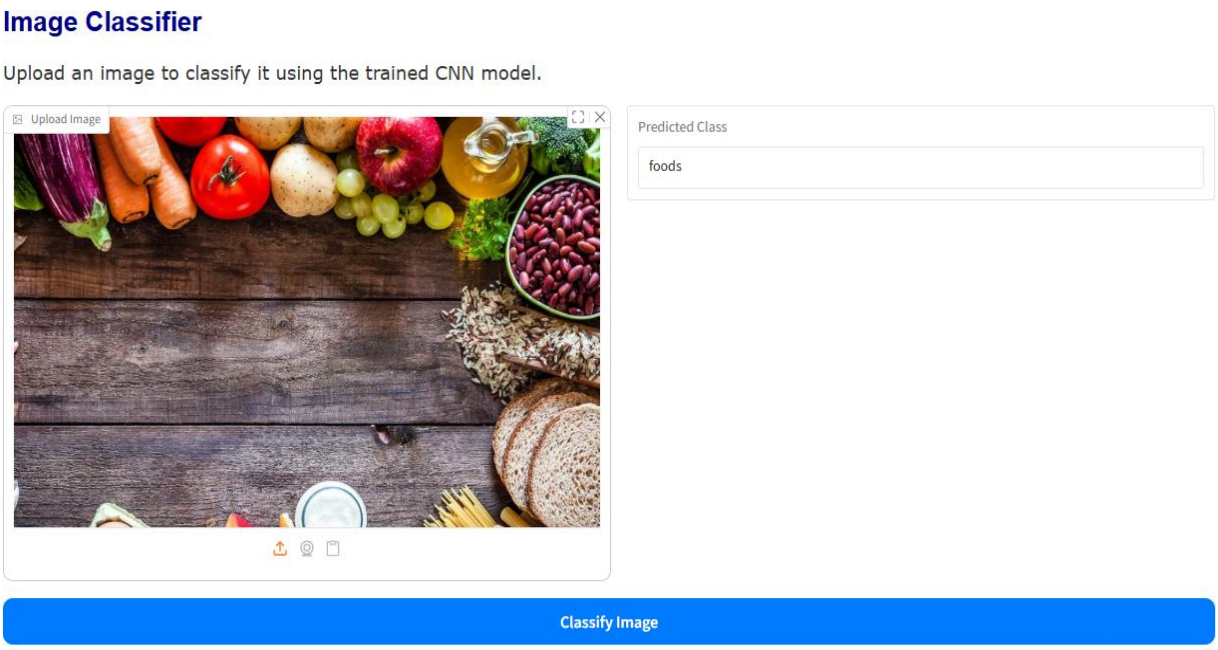

Figure53 Classified image

Displaying the Classification Result

The predicted class is displayed in a result box on the interface, allowing the user to immediately see the outcome of the classification.

8. Conclusion

The objective of this chapter was to present a comparative study of several image classification methods on the Corel-1K and Kimia Path960 datasets.

The study focused on five classifiers: KNN, SVM, Naive Bayes, Decision Tree, and Random Forest. For each classifier, five feature extraction methods were applied: LBP, Haralick, HSV, Color Moments, and Fourier Descriptors. Additionally, a CNN classifier was also implemented and evaluated.

The results of the study showed that:

- The performance of the method depends on the complexity and richness of the semantic content.
- The CNN classifier achieved good results but remains sensitive to the number of epochs and the size of the training dataset.

- The best method depends on the organization of the dataset, with HSV generally yielding the best results, followed by Color Moments.

In conclusion, this study has provided a deeper understanding of the performance of certain methods, allowing for the extraction of valuable insights. The results obtained can be useful for recognizing the effectiveness and power of each method, depending on the specific requirements and criteria.

General Conclusion

Image classification has proven to be a relevant gateway for better understanding the mechanisms of Data Mining, particularly within the subdomain of Image Mining, which opens a wide range of applications in various fields such as object recognition, anomaly detection, content-based image retrieval, and medical image analysis.

In this thesis, we implemented several machine learning classifiers namely KNN, SVM, Naive Bayes, Decision Tree, and Random Forest combined with various "hand-crafted" image descriptors and different color spaces. We also used a deep learning classifier (CNN) for comparison. The KNN and CNN classifiers were tested on two distinct image datasets: Corel 1k, a general-purpose image dataset rich in visual concepts, and Kimia Path960, a dataset specialized in histological medical images.

The results led to several key findings:

1. All parameters tested (classifier type, color space, used descriptors, and dataset arrangement) significantly influenced the classification performance.

2. CNNs proved to be much more effective on images rich in concepts (Corel 1k), while their performance was less effective on medical images with fewer visual concepts (Kimia).

3. There is variability in how easily images can be classified, depending on their richness or lack of conceptual content.

In conclusion, image classification has established itself as a powerful tool for analysis and knowledge extraction from digital images. The choice of technique and approach depends on the specific requirements of each application domain.

Through this work, we gained important insights:

1. We successfully applied image classification in the context of Data Mining, focusing on its image mining aspect.

2. We became aware of the importance of color spaces in image mining tasks, particularly for image retrieval, classification, and anomaly detection.

3. We better understood the role of hand-crafted descriptors in extracting relevant features for image mining.

4. We explored a variety of classifiers, from simpler ones to more advanced methods, allowing us to compare the effectiveness of classic machine learning methods with that of deep learning.

This work thus serves as a solid foundation for future research in image classification, particularly in sensitive and demanding fields such as medical imaging.

References

[1] IBM, 2023. https://www.ibm.com/think/topics/datawarehouse.

[2] PingCAP, 2023. https://www.pingcap.com/article/real-time-data-warehouse-benefits/.

[3] 365 Data Science, 2023. https://365datascience.com/trending/data-cube/.

[4] Agrawal, R., & Srikant, R., 1994. *Fast Algorithms for Mining Association Rules.* In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, pp. 487–499.

[5] Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P., 1996. *From Data Mining to Knowledge Discovery in Databases.* AI Magazine, 17(3), pp. 37–54.

[6] GeeksforGeeks. KDD Process in Data Mining. https://www.geeksforgeeks.org/kdd-process-in-data-mining, s.d.

[7] Tableau, **2023.** *Data Cleaning: Definition, Benefits, And How-To.* https://www.tableau.com/learn/articles/what-is-data-cleaning.

[8] Susanto, ***A., & Meiryani, 2019.*** *Functions, Processes, Stages and Application of Data Mining.* International Journal of Scientific & Technology Research, 8(7), pp. 120-124. ISSN 2277-8616.

[9] F. Bougamouza, Deep Learning , document de cours, Univ. 20 Août 1955 Skikda, Dépt. Informatique, Skikda, Algérie, 2023–2024.

[10] Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.

[11] Sonka, M., Hlavac, V., & Boyle, R., 2014. *Image Processing, Analysis, and Machine Vision (4th ed.).* Cengage Learning.

[12] https://en.wikipedia.org/wiki/Alpha_compositing. (accessed 2025)

[13] http://users.iit.demokritos.gr/~bgat/HDIBCO2014/benchmark. (accessed 2023)

[14] https://rosettacode.org/wiki/Grayscale_image. (accessed 2025)

[15] https://medium.com/@abethcrane/color-me-knowledgeable-e69168241c7c. (accessed 2025)

[16] Zubair, M., & Alo, A., 2016. *Grey Level Co-occurrence Matrix (GLCM) Based Second Order Statistics for Image Texture Analysis.*

[17] https://www.istockphoto.com/photos/wood-texture. (accessed 2025)

[18] https://fr.vecteezy.com/art-vectoriel/15370127-vue-de-dessus-de-fond-de-sable-texture-du-desert-ou-de-la-plage). (accessed 2025)

[19] Sedaghatjoo, Z., Hosseinzadeh, H., & Bigham, B. S., 2024. *Local Binary Pattern (LBP) Optimization for Feature Extraction.* arXiv preprint arXiv:2407.18665. https://arxiv.org/abs/2407.18665.

[20] Retrieved May 2025 https://www.researchgate.net/figure/.

[21] Harbi, W., & Nedjar, S. (2024). *Classification d'Images dans le contexte de l'Image Mining : Une étude comparative selon plusieurs critères* (Mémoire de master, Juin 2024). Supervised by Pr. Boucheham Bachir.

[22] Gonzalez, R. C., & Woods, R. E., 2002. *Digital Image Processing (2nd ed.).* Prentice Hall.

[23] RGB model representation [Image]. (n.d.). In *ResearchGate*. Retrieved May 20, 2025, from https://www.researchgate.net/figure/RGB-model-representation_fig1_239814963.

[24] https://www.testprint.net/exploring-color-models-hsl-hsv-and-lab/(accessed 2025)

[25] https://www.linshangtech.com/tech/lab-color-model-color-meter-tech1432.html. (accessed 2025)

[26] https://www.researchgate.net/figure/YCbCr-Color-Space-4_fig3_357594587. (accessed 2025)

[27] Rabie, T., Baziyad, M., Sani, R., Bonny, T., & Fareh, R. (2024). *Color Histogram Contouring: A New Training-Less Approach to Object Detection*. Electronics, 13(13), 2522. https://doi.org/10.3390/electronics13132522.

[28] https://www.researchgate.net/figure/Exemple-dhistogramme-dune-image-couleur-Une-des-premieres-approches-decrivant_fig4_331929280. (accessed 2025)

[29] Keen, N., 2005. *Color Moments*. Retrieved from https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0405/KEEN/av_as2_nkeen.pdf.

[30] https://en.wikipedia.org/wiki/Color_moments#Overview. (accessed 2025)

[31] Zhang, D., & Lu, G., 2004. *Review of shape representation and description techniques*. Pattern Recognition, 37(1), pp.1–19. https://doi.org/10.1016/j.patcog.2003.07.008.

[32] Srinivasa Rao, Ch., Srinivas Kumar, S., & Chandra Mohan, B., 2010. *Content Based Image Retrieval Using Exact Legendre Moments and Support Vector Machine*. arXiv preprint arXiv:1005.5437. https://arxiv.org/abs/1005.5437.

[33] Martínez-Muñoz, M., Rodríguez-Quiñonez, J.A., Nájera, M., & Villalobos-Vázquez, O.O., 2024. *Fourier Features and Machine Learning for Contour Profile Inspection in CNC Milling Parts: A Novel Intelligent Inspection Method (NIIM)*. Applied Sciences, 14(18), 8144. https://doi.org/10.3390/app14188144.

[34] Gower, J.C., & Warrens, M.J., 2017. *Similarity, Dissimilarity, and Distance Measures*. In Wiley StatsRef: Statistics Reference Online. Wiley. https://doi.org/10.1002/9781118445112.stat02470.pub2.

[35] https://www.sciencedirect.com. (accessed 2025)

[36] Lance, G. N., & Williams, W. T., 1967. *Mixed-data classificatory programs I. Agglomerative Systems*. Australian Computer Journal, 1(1), 15–20.

[37] https://en.wikipedia.org/wiki/Similarity_measure.

[38] Ribeiro, A., & de Mello, R. A., 2020. *Image Similarity Techniques: A Survey*. Pattern Recognition and Image Analysis, 30(3), 611–624. https://doi.org/10.1134/S1054661820030132.

[39] Hsu, W., Lee, M. L., & Zhang, J. (2002). Image mining: Trends and developments. *Journal of Intelligent Information Systems, 19*(1), 7–23. https://doi.org/10.1023/A:1011910500527.

[40] Oliinyk, Y., Kapshuk, M., & Oliinyk, L. (2024). Software for anomaly detection in MRI images. *Informatics and Data-Driven Medicine: Proceedings of the 7th International Conference IDDM 2024*, 3892, 160–170. CEUR Workshop Proceedings.

[41] Fadaei, S., Rashno, A., & Rashno, E. (2019). *Content-based image retrieval speedup*. arXiv. https://arxiv.org/abs/1911.11379.

[42] Aznag, K., Datsi, T., El Oirrak, A., & El Bachari, E. (2020). Binary image description using frequent itemsets. *Journal of Big Data, 7*(1), 32. **https://doi.org/10.1186/s40537-020-00307-8.**

[43] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems (Vol. 25, pp. 1097–1105). Curran Associates, Inc. https://papers.nips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[44] Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). *A survey of the recent architectures of deep convolutional neural networks*. Artificial Intelligence Review, 53(8), 5455–5516. https://doi.org/10.1007/s10462-020-09825-6.

[45] Shrestha, A., & Mahmood, A. (2019). https://doi.org/10.1109/ACCESS.2019.2912200.

[46] Zhang, J., & Lu, G. (2008). *A review of image preprocessing techniques for classification*. In *Proceedings of the International Conference on Image and Graphics* (pp. 229–234). Springer.

[47] Zhang, D., & Li, X. (2009). *Image feature extraction and image classification techniques*. In *Proceedings of the International Conference on Information and Automation* (pp. 81–86).

[48] Powers, D. M. W. (2011). *Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation*. Journal of Machine Learning Technologies, 2(1), 37–63.

[49] MathWorks. (n.d.). *Support Vector Machines for Binary Classification*. https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html.

[50] https://www.nomidl.com/machine-learning/support-vector-machine-algorithm-for-machine-learning.

[51] D. Saini, T. Chand, D. K. Chouhan, and M. Prakash, "A comparative analysis of automatic classification and grading methods for knee osteoarthritis focusing on X-ray images.

[52] Murphy, K. P. (2012).
Machine Learning: A Probabilistic Perspective. MIT Press.

[53] Zhang, H. (2004).
The optimality of naive Bayes. Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference (FLAIRS), 1–6.

[54] Manning, C. D., Raghavan, P., & Schütze, H. (2008).
Introduction to Information Retrieval. Cambridge University Press.

[55] Mitchell, T. M. (1997).
Machine Learning. McGraw-Hill.

[56] Jain, A.K., Duin, R.P.W., & Mao, J. (2000).
Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(1), 4–37. https://doi.org/10.1109/34.824819.

[57] Vishwakarma, R., & Kumar, J. J. (2023).
K-Nearest Neighbours. *CS456 - Machine Learning*, Spring 2023, School of Computer Sciences, National Institute of Science Education and Research, Homi Bhabha National Institute. February 19, 2023.

[58] The Startup. (2020, May 3). *K-Nearest Neighbor*. Medium. Retrieved from https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4.

[59] Han, J., Kamber, M., & Pei, J. (2012).
Data Mining: Concepts and Techniques (3rd ed.). San Francisco, CA, USA: Morgan Kaufmann.

[60] Udacity, Inc. (2014). *ID3 Algorithm for Decision Trees*. https://s3-us-west-2.amazonaws.com/gae-supplemental-media/id3-algorithm-for-decision-treespdf/ID3-Algorithm-for-Decision-Trees.pdf.

[61] Rhodes, D. (n.d.). *Decision Trees*. Juniata College. Retrieved from https://jcsites.juniata.edu/faculty/rhodes/ida/decisionTrees.html.

[62] Ren, Y., Zhu, X., Bai, K., & Zhang, R. (2024). A new random forest ensemble of intuitionistic fuzzy decision trees. *arXiv preprint arXiv:2403.07363*. https://arxiv.org/abs/2403.07363.

[63] Altman, N., & Krzywinski, M. (2017**).** Ensemble methods: bagging and random forests. *Nature Methods, 14*(10), 933–934. https://doi.org/10.1038/nmeth.4426.

[64] Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324.

[65] Nasief, H., Zheng, C., Schott, D., Hall, W., Chakkungal, S., Tsai, S., Erickson, B., & Li, X. (2020). Improving treatment response prediction for chemoradiation therapy of pancreatic cancer using a combination of delta-radiomics and the clinical biomarker CA19-9. *Frontiers in Oncology, 10*, 1–10. https://doi.org/10.3389/fonc.2020.00001.

[66] The Impact of Dimensionality Reduction Techniques on Machine Learning Algorithm Efficiency**.** (2024). *ResearchGate*. https://www.researchgate.net/publication/380123456_The_Impact_of_Dimensionality_Reduction_Techniques_on_Machine_Learning_Algorithm_Efficiency.

[67] Lyu, J., Zhou, K., & Zhong, Y. (2025). *A Statistical Theory of Overfitting for Imbalanced Classification*. arXiv**. https://arxiv.org/abs/2502.11323.**

[68] LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539].

[69] Raj, R., & Kos, A. (2023). An improved human activity recognition technique based on convolutional neural network. *Scientific Reports*, 13, Article 22581. https://doi.org/10.1038/s41598-023-49739-1.