RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Electronique

End of study project dissertation for obtaining the State Engineer's degree in Electronics

---

Study Of Iris Recognition Using Human Visual System Based Methods

---

BENNEFISSA Mahmoud

Under the supervision of
Pr. HAMMAMI and **Dr. ALLAM Fatima Zohra**

Publicly presented and defended on the 15^{th} of October, 2024

Jury members :

President	Pr. Bousbia-Salah Hicham	ENP
Supervisor	Pr. HAMMAMI Latifa	ENP
Co-supervisor	Dr. ALLAM Fatima Zohra	ENP
Examiner	Dr. BOUADJENEK Nesrine	ENP

ENP 2024

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Electronique

End of study project dissertation for obtaining the State Engineer's degree in Electronics

Study Of Iris Recognition Using Human Visual System Based Methods

BENNEFISSA Mahmoud

Under the supervision of
Pr. HAMMAMI Latifa and **Dr. ALLAM Fatima Zohra**

Publicly presented and defended on the 15^{th} of October, 2024

Jury members :

President	Pr. Bousbia-Salah Hicham	ENP
Supervisor	Pr. HAMMAMI Latifa	ENP
Co-supervisor	Dr. ALLAM Fatima Zohra	ENP
Examiner	Dr. BOUADJENEK Nesrine	ENP

ENP 2024

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Electronique

Mémoire de projet de fin d'études pour l'obtention du diplôme d'Ingénieur d'état en Electronique

Etude de la Reconnaissance d'Iris en utilisant le Système Visuel Humain

BENNEFISSA Mahmoud

Supervisé par
Pr. HAMMAMI Latifa et **Dr. ALLAM Fatima Zohra**

Présenté et soutenu publiquement le 15 Octobre 2024

Membres du jury :

Président	Pr. Bousbia-Salah Hicham	ENP
Encadrant	Pr. HAMMAMI Latifa	ENP
Co-encadrant	Dr. ALLAM Fatima Zohra	ENP
Examinatrice	Dr. BOUDJENEK Nesrine	ENP

ENP 2024

ملخص :

تعتبر تقنية التعرف على قزحية العين باستخدام أساليب مستوحاة من نظام الرؤية البشري نهجًا متقدمًا في مجال الهوية البيومترية. من خلال تحليل المراحل الأساسية مثل الكشف، والتطبيع، واستخراج الميزات، والتصنيف، تُظهر هذه الدراسة كيف يمكن لتقنيات مثل كشف الحواف، ومرشحات غابور، وتحويلات الموجات أن تعزز بشكل كبير دقة وموثوقية التعرف. بالإضافة إلى ذلك، توفر دراسة تقنية FPGA مسارًا نحو تنفيذ فعّال للأجهزة. تسهم النتائج في تقدم أنظمة التعرف على قزحية العين من خلال دمج الأطر النظرية مع التطبيقات العملية.

الكلمات المفتاحية: التعرف على قزحية العين، نظام الرؤية البشري، معالجة الصور، مرشحات غابور، FPGA، استخراج الميزات.

Résumé

La reconnaissance de l'iris utilisant des méthodes inspirées du système visuel humain représente une approche de pointe dans l'identification biométrique. En analysant les étapes clés telles que la détection, la normalisation, l'extraction de caractéristiques et la classification, cette étude démontre comment des techniques telles que la détection des contours, les filtres de Gabor et les transformations en ondelettes peuvent améliorer de manière significative la précision et la robustesse de la reconnaissance. De plus, l'exploration de la technologie FPGA offre une voie pour une mise en œuvre matérielle efficace. Les résultats contribuent à l'avancement des systèmes de reconnaissance de l'iris en intégrant des cadres théoriques avec des applications pratiques.

Mots clés : Reconnaissance de l'iris, système visuel humain, traitement d'image, filtres de Gabor, FPGA, extraction de caractéristiques.

Abstract

Iris recognition using methods inspired by the human visual system represents a cutting-edge approach in biometric identification. By analyzing key stages such as detection, normalization, feature extraction, and classification, this study demonstrates how techniques like edge detection, Gabor filters, and wavelet transforms can significantly enhance recognition accuracy and robustness. Additionally, the exploration of FPGA technology provides a pathway for efficient hardware implementation. The findings contribute to advancing iris recognition systems by integrating theoretical frameworks with practical applications.

Keywords : Iris recognition, human visual system, image processing, Gabor filters, FPGA, feature extraction.

Dedications

I dedicate this work

In the first place to my dear parents for their unwavering support throughout my entire educational journey.

To my brother and my whole family for their continuous support.

To Blida Qoran friends group with whom I made very special and unforgettable memories during my whole life, not forgetting their guide which always proved to be important and precious.

To 7a9ou my bestest friend, to My genshin friends Amine Rosso and Said Griezo, to the Happy Farm group which without them I wouldn't pass my 2 years of Prepa, from Sabri to Ouanis to Soundous to Nourhane to Ivan and Aizen, Also the best Group ever Vnivere who made the rest of the years Memorable and really enjoyable without fogetting their emotional help, to the other Polytech friends such as Rayan and his 3 personas, Ridha Mazari and everyone who made my journey in Polytech memorable.

To Bahaa Boulehya, Mohamed Chadi, Abdeslam Jijli, Hamza. To Wassim Ferrad, Ascem Taijou, Said Greizo again with the assists Group. To Sora, Asta, Moony and who helped me immensely making this thesis.

To all those who have directly or indirectly contributed to the realization of this project.

Thank you sincerely to everyone.

Mahmoud

Acknowledgments

I would like to express my sincere gratitude to everyone who has supported me throughout my research and the writing of this thesis.

First and foremost, I am deeply grateful to my supervisors HAMMAMI Latifa,Dr. ALLAM Fatima Zohra, for their invaluable guidance and continuous support during my research. Their insightful feedback and advice have greatly improved the quality of my work.

I also extend my sincere thanks to Pr. Bousbia-Salah Hicham for accepting to be president of the jury and also Dr. BOUDJENEK Nesrine for taking the time and effort in examining this thesis.

Let's not forget about the best class I have ever studied in ever, and the best people I could have known through a school or a university, ELN people from my class with frfr, wissou, mchimcha,les mehdis, rahal, foutia to the senpais like walid zegaoui, Maroi and all ELN people I have ever known. Without them the last 3 years and even the thesis wouldn't have been like this.

Finally, I would like to thank Boutiara Abdelhak for his precious counsel, irreplaceable support, as well as the highest form of selflessness. Without him this thesis would not have been possible.

Table of Contents

List of Figures

List of Tables

List of Acronyms

- **AI** : Artificial Intelligence
- **DL** : Deep Learning
- **CNN** : Convolutional Neural Network
- **VHDL** : Very High Speed Integrated Circuit Fast Hardware Discription Languqge
- **FPGA** : Feild Programmable Gate Array
- **PYNQ** : Python Productivity for ZYNQ
- **HDL** : Hardware Description Language
- **HVS** : Human Visual System
- **DWT** : Discrete Wavelete Transform
- **HOG** : Histogram of Oriented Gradients
- **DCT** : Discrete Cosine Transform
- **GF** : Gabor Filters
- **SVM** : Support Vector Machine
- **KNN** : K-Nearest Neiboghrs
- **GPU** : Graphics Processing Unit
- **CPU** : Central Processing Unit
- **LoG** : Laplacien of Gaussien
- **HT** : Hough Transform
- **PCA** : Princple Components Analysis
- **CSF** : Contrast Sensitivity Function

General introduction

Biometrics, as a field dedicated to the measurement and analysis of individuals' unique biological characteristics, has revolutionized methods of identification and authentication. In an increasingly interconnected and digital world, the security of access systems has become paramount. Among the various biometric modalities, iris recognition stands out for its ability to offer precise and reliable identification, thanks to the complexity and stability of iris patterns. Unlike other biometric traits, iris patterns are not only unique but remain unchanged over time, making them an ideal choice for modern authentication systems.

Recent technological advancements have enabled researchers to overcome several challenges associated with iris recognition, particularly regarding variations in lighting and environmental conditions. Current research focuses on integrating artificial intelligence and machine learning methods to enhance the performance of biometric systems. By adopting techniques inspired by the human visual system, it is possible to optimize image analysis processes, thereby increasing the accuracy and robustness of recognition systems.

This thesis concentrates on the study and implementation of iris recognition using methods based on the human visual system. We analyze the essential steps involved, from detection to identification, highlighting the applied image processing techniques, such as edge detection, segmentation, normalization, and feature extraction. The use of human visual system-inspired models, such as Gabor filters and wavelet transforms, significantly improves the efficiency of the recognition process.

Finally, we explore the hardware implementation methods, particularly the use of FPGA technology, to leverage parallel processing capabilities and high performance. This work aims to contribute to the advancement of iris recognition systems by combining theoretical approaches with practical applications, while paving the way for future research in the field.

This thesis is structured as follows:

- Chapter 1: Overview of the Human Visual System and Iris Recognition.
- Chapter 2: State of the Art - Literature Review.
- Chapter 3: Terminology and Algorithms for Human Visual System.
- Chapter 4: Proposed Approach and Results .
- General conclusion – Summarizes the findings, discusses their implications, and suggests directions for future research.

Chapter 1

Overview of the Human Visual System and Iris Recognition

1.1 Introduction

This chapter provides an overview of biometric recognition systems, focusing on iris recognition and its implementation. This chapter explores the key principles that make biometric methods reliable and secure, while highlighting the unique features and challenges of using the iris for identification. The chapter also touches on the Human Visual System (HVS) from both psychological and technological perspectives, as it plays a significant role in recognition processes. Lastly, it introduces the hardware considerations for implementing such systems, with a focus on FPGA technology, which will be the platform used in this thesis.

1.2 Recognition Systems

In this section, we will provide an overview of several biometric recognition systems and identity verification techniques, with a particular focus on iris biometrics.

1.2.1 Biometric Recognition

Biometrics is the science that studies methods for identity verification or authentication, which are used to distinguish between individuals based on the recognition of their biological characteristics, wether it is physiological or behavioral. It serves as a powerful means of identity verification in some applications when correctly implemented. Furthermore, when combined with other verification techniques (such as keys and passwords), a certain level of security can be achieved [3]. This technology utilizes various traits such as fingerprints, facial features, iris patterns, and voice.

The reasons for choosing biometric systems are :

- **Security :** It improves security by producing a more trustworthy way to verify the identity of, for example in law enforcement, border control or access control to secure facilities.

- **Fraud Prevention :** Client is authorized on a few basic things and he can spare time for another correct work, which isn't included in his busy schedule so Broader convenience: As you know people rarely keep spares standing around, but they do have their phone with them.

- **Ease of use :** Biometric systems do away with the need for passwords or actual tokens, meaning that users can authenticate themselves rapidly and without any hassle.

- **Scalability :** Since they can manage thousands of biometric feature sets, these systems are ideal for applications that have to identify many individuals en masse (e.g. airports and other high-traffic, large-company settings).

- **Higher accuracy :** Biometric systems can provide better performance than the traditional method of identification, which means reducing error rates of false positives in identity verification.

1.2.2 Biometric System

A biometric system is a pattern recognition system that operates by acquiring biometric data from an individual, extracting a set of features from the acquired data, comparing this set of features to the template stored in database and accepts or rejects the demanded identity. There are different modes in which a biometric system can function based on its application context[4]:

1- Verification : It does this by matching the biometric data captured to its own database to verify that the person it is scanning is who he claims to be. The system works like the image process, only now being a person trying to get itself identified thus it will declare an identity with its own PIN (personal identification number) one username, or an ID card etc. In that case, the system compares them one-one to see if it is true or not.

2- Recognition: The system searches across all the templates in a database and then assigns the input template to one of them. Therefore, the system peform various comparisons to establish the individual's identity

A representation of biometric recognition operating scheme is illustrated in the figure 1.1

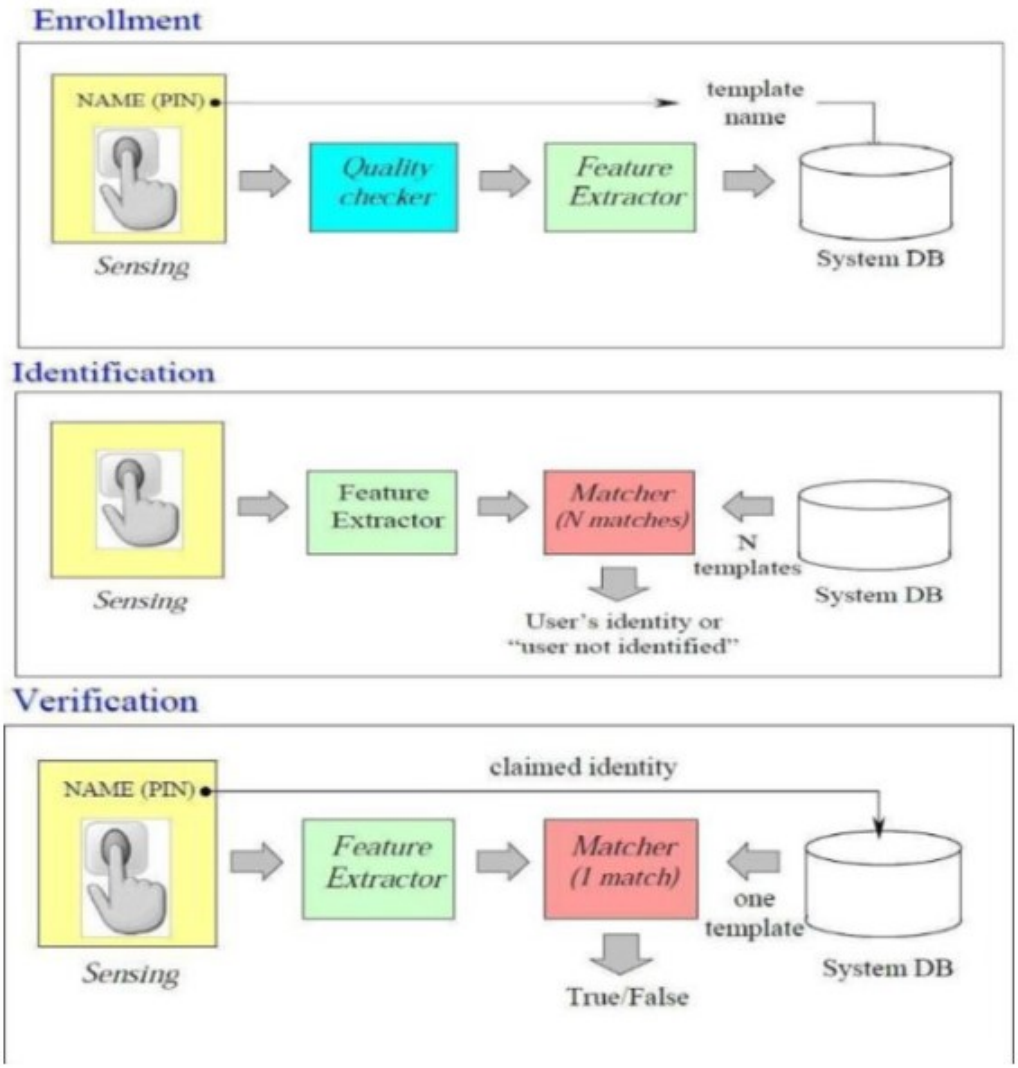


Figure 1.1: Synoptic Diagram of Bimetric Systems[1]

1.2.3 Condition for a Biometric Recognition Method

Automatic biometric systems must identify an individual or verify their identity using measurements of human body characteristics. These characteristics must have certain intrinsic qualities to allow for the development of reliable and robust authentication systems. In theory, any human characteristic can be used for identity verification, as long as it meets the following desirable (ideal) requirements :

- **Universality:** Every person should have the characteristic.
- **Uniqueness:** The characteristic should ensure that no two individuals have the same traits.
- **Permanence:** The characteristic should remain constant and unchanging over time.
- **Measurability:** It should be possible to collect a sample of the characteristic and analyze it in a quantitative manner.

In practice, there are additional important requirements for the identification and authentication phases:

- **Performance:** This refers to achieving both accurate verification/identification and efficient use of resources to meet acceptable operational conditions.
- **Robustness:** The system's ability to maintain accuracy despite the influence of operational or environmental factors (such as channel noise, deformations, and other distortions).
- **Acceptability:** This refers to how willing users are to accept the biometric system and use it.
- **Anti-spoofing Capability:** The system's resistance to deception techniques, ensuring that an individual cannot fraudulently pass their biometric data to someone else for misuse.

These requirements apply to both identification and verification phases. The closer a biometric modality comes to meeting these requirements, the closer the system will be to perfection. However, in practice, no biometric modality perfectly fulfills all these ideal conditions [3].

1.2.4 Different Types of Biometric Recognition

Biometric systems are increasingly integral to identity verification across various sectors, utilizing unique physiological and behavioral traits of individuals. These systems can be categorized into several types as indicated in figure 1.2, each with distinct characteristics and applications.

1. **Physiological Biometrics :** These systems identify individuals based on unique physical characteristics.
    - **Fingerprint Recognition :** This is one of the most widely used biometric methods. It captures the unique patterns of ridges and valleys on a person's fingers. The system analyzes minutiae points—specific features in the fingerprint pattern—and creates a template for matching during authentication.
    - **Facial Recognition :** Utilizing advanced algorithms, facial recognition technology identifies individuals by analyzing facial features from images or video feeds. This method is commonly used in security systems and personal devices, allowing for quick and contactless verification.
    - **Iris Recognition :** This method involves scanning the intricate patterns of the iris, which are unique to each individual. Iris recognition is highly accurate and is often employed in secure access control systems.
    - **Retina Scanning :** Similar to iris recognition, retina scanning examines the unique patterns of blood vessels in the retina at the back of the eye. This method is less common due to its invasive nature but offers high accuracy.
    - **DNA Matching :** DNA biometrics analyze genetic material to establish identity. While highly accurate, this method is typically reserved for forensic applications due to the complexity and time required for analysis.

2. **Behavioral Biometrics :** These systems identify individuals based on their behavioral patterns.
    - **Voice Recognition :** This technology analyzes vocal characteristics such as pitch, tone, and speech rhythm. It is commonly used in call centers and security applications where voice authentication can streamline user access.
    - **Keystroke Dynamics :** This method involves analyzing the unique patterns in how an individual types, including speed and rhythm. It can provide continuous authentication without requiring active participation from the user.
    - **Signature Dynamics :** This system examines the way a person signs their name, including pressure and speed variations during the signing process. It is often used in banking and legal applications to verify identity.

In real life, there exist a multimodal systems that combine multiple biometric identifiers to enhance accuracy and security. For example, a system might use both fingerprint scanning and facial recognition to authenticate a user. This approach mitigates the limitations of unimodal systems, such as false rejections or vulnerabilities due to changes in a single biometric trait over time.

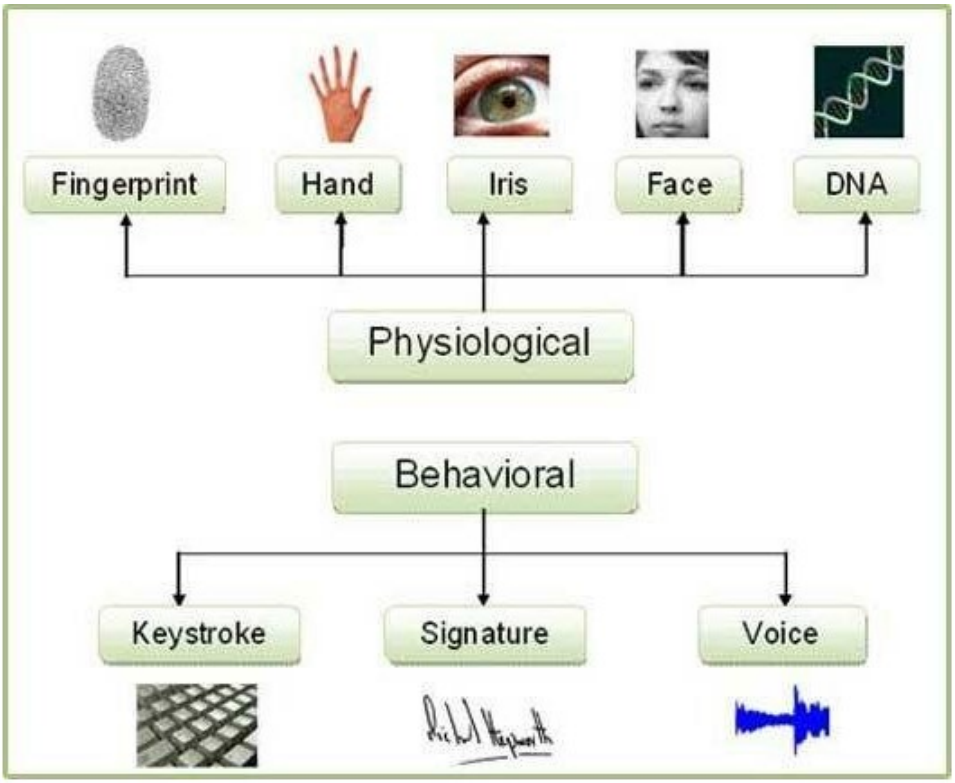


Figure 1.2: Different types of Biometric Systems[2]

1.2.5 Characterisation of a Biometric System

A typical biometric system can be described through four main modules:

- **Capture Module :** The capture module is responsible for acquiring the biometric data of an individual. This can include devices such as cameras, fingerprint scanners, or security cameras.

- **Feature Extraction Module :** The feature extraction module takes the biometric data obtained from the capture module and isolates only the relevant information to create a new representation of the data. Ideally, this representation should be unique to each individual and relatively invariant to intraclass variations.
- **Matching Module :** The matching module compares the set of extracted features with the stored model in the system's database. Assesses the degree of similarity (or difference) between the two sets of data.
- **Decision Module :** The decision module verifies the identity claimed by a user or determines an individual's identity based on the degree of similarity between the extracted features and the stored model(s). In summary, these four components work together to ensure accurate and reliable biometric identification, from data acquisition to final identity verification.

1.3 Iris Recognition

The iris is a circular membrane located at the front of the eye and plays a crucial role in regulating the amount of light that enters. In its center is the pupil, a small aperture through which light passes to reach the retina. The iris adjusts its size in response to lighting conditions; for example, in bright environments, it constricts to limit light entry, while in darker settings, it dilates to allow more light in.[5]

The intricate texture of the iris is composed of various elements, including ligament arcs, crypts, ridges, grooves, and collarettes. This complex arrangement and the specific shapes of these components contribute to the distinctive appearance of the iris.

1.3.1 Why Iris Recognition

The iris is unique among internal organs which are well protected inside the eye, because it can also be seen from outside the body, as it is protected by the transparent cornea. This accessibility makes it easier to acquire its texture compared to other internal structures such as the retina. The colorful pattern circling the pupil, on the iris, is complex and distinct, that is why it has spurred interest from researchers and ophthalmologists in developing iris recognition systems since the 1930s. In fact, a patent was filed in 1986 asserting that the irises of different individuals are not identical[5], highlighting the potential for iris patterns to be used in biometric identification. Secondly, Iris patterns remain constant throughout a person's life, which has formed the basis on which iris recognition rests as a very accurate biometric technique boasting of an exceptionally low False Match Rate.

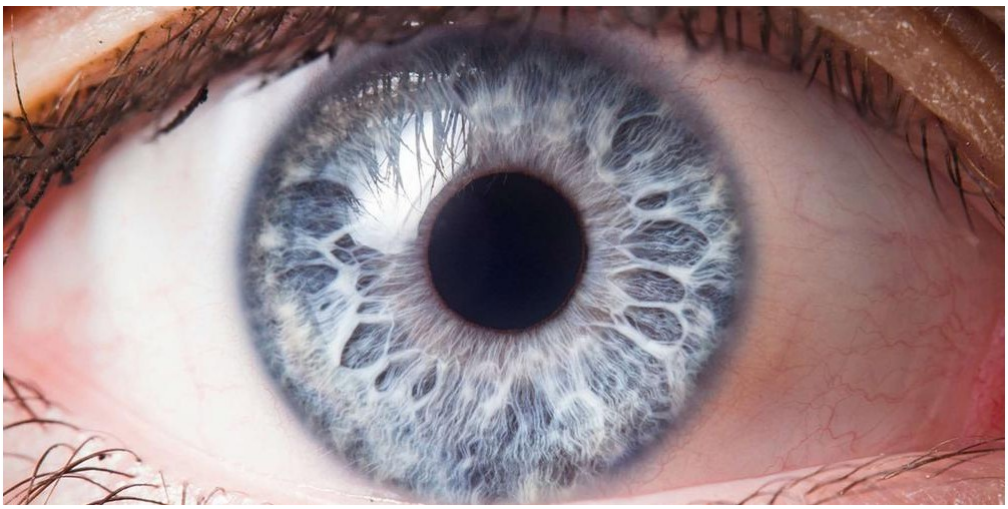


Figure 1.3: Iris Textures

1.3.2 Anatomy of the Iris

The iris plays a critical role in regulating pupil dilation[6], acting as a circular diaphragm that automatically adjusts based on the amount of light received. In low light conditions, the pupil expands, which can lead to a less sharp image on the retina, a phenomenon known as night myopia. The iris is also responsible for eye color, which is determined by the thickness and concentration of pigmented layers; a thicker layer with higher melanin concentration results in darker eye colors. Its nutrition is provided by the aqueous humor surrounding it, along with several small arterioles. The diameter of the iris is altered by two types of muscles: the dilator muscle, which contracts to widen the pupil, and the sphincter muscle, which decreases the pupil's diameter. Understanding the anatomy and physiology of the iris is essential to grasp how our visual system functions as a whole. The figure 1.4 represents the regulation of pupil by the iris in bring and dim light conditions.

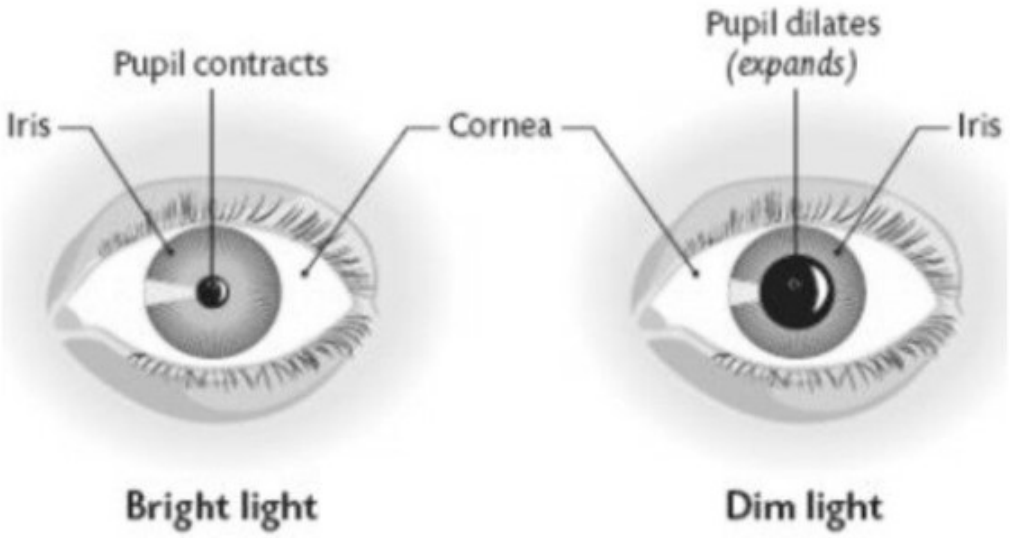


Figure 1.4: Variation of pupil in bright and dim light conditions

1.3.3 Difficulties in Iris Recognition System

Iris biometrics faces several challenges in creating a reliable recognition system. The main difficulty of an iris recognition is its acquisition system. The iris, visible due to the reflective cornea, can present complications from reflections on the pupil and the iris during acquisition. In addition, occlusion noise can obscure the texture of the iris, caused by the eyelids and eyelashes, reflections, lenses etc. which can cover significant portions of the iris. The presence of glasses or contact lenses can also introduce further reflections. Blurring noise is another concern, as capturing the small iris from a distance requires sensitive lenses, making the individual's distance from the camera critical; otherwise, images can appear blurry. Motion blur can also occur from rapid eye movements or changes in pupil size. In addition, the texture of the iris varies between individuals, affecting the accuracy of recognition. Finally, the response of the iris muscle to light can distort its texture, leading to unpredictable and non-linear deformations.

1.3.4 Architecture of Iris Recognition System

Like the other recognition technologies, the iris recognition system is structured around two primary phases: iris detection and iris identification (or matching). The first phase covers several steps including preprocessing, edge detection, segmentation, and normalization. In the second phase, the focus shifts to feature extraction and classification.

1.3.4.1 The Detection Phase

- **Preprocessing :** involves essential steps to prepare the image for subsequent analysis. This phase starts with image acquisition, during which high-quality facial images are captured to ensure precision. Various image enhancement techniques are then utilized to improve the overall quality of the captured images.

- **Iris Detection and Segmentation :** Once the preprocessing phase is complete, the next essential step is edge detection. This process is crucial for highlighting the boundaries of the iris. Following that, the localization of the iris boundaries occurs, isolating the iris region while discarding other parts of the image.

- **Normalization :** The iris is a disc that is pierced in the center by a smaller disc known as the pupil. The two circles formed by the boundaries of the iris and the sclera, as well as the boundaries between the pupil and the iris, are not perfectly concentric. Additionally, due to the contractions and dilations of the iris, along with varying distances between individuals and the camera, the size of the iris disc is not always uniform, therefore we transform the circular iris region into a rectangular shape, making it suitable for further processing.

1.3.4.2 The identification phase

- **Features Extraction :** which involves extracting the most important features from the resulting rectangle obtained during the normalization stage. It is important to note that both feature extraction and feature selection are dimension reduction methods used for this purpose.

The feature extraction aims to identify and extract the most relevant features from the input data while reducing its dimensionality. However, feature selection involves choosing a subset of the original features to decrease dimensionality. Both techniques aim to enhance the performance of the model.

- **Classification :** Once the most significant features are extracted, we proceed to the final stage of our system, which is the Classification. Classification involves comparing the encoded iris pattern (generated during feature extraction) to stored templates in a database. The system assesses the similarity between the patterns to determine if there is a match, thereby verifying or identifying the individual's identity. This step is crucial for making the final decision on recognition.

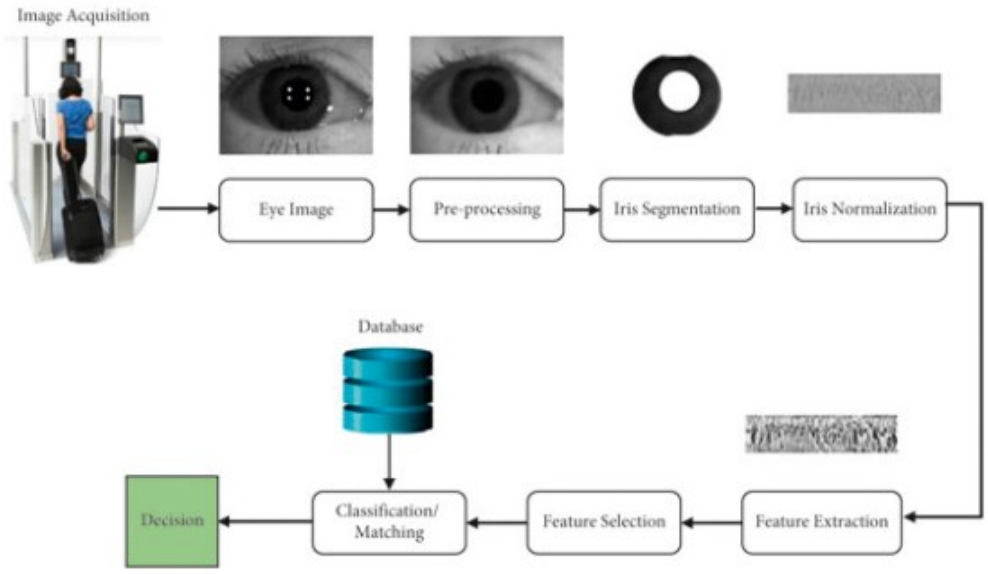


Figure 1.5: Iris Recognition System Architecture

1.4 Human Visual System

1.4.1 Definition of HVS

The HVS refers to the biological and psychological processes that enable humans to interpret visual stimuli. HVS encompasses all the biological processes involved in seeing. It includes the eyes, the optic nerve, and the brain regions responsible for visual perception. The HVS detects light, processes visual information, and interprets this data to create an understanding of the world. It involves complex processes like color vision, depth perception, motion detection, and recognition of objects and patterns.

In image processing, understanding the HVS allows for the development of algorithms that align with human perception, making visual information more accessible and efficient. For instance, methods derived from the HVS can optimize image quality while reducing data size. It refers to how image-processing techniques take advantage of the characteristics of human vision. Algorithms are designed to prioritize details that the HVS is more sensitive to (such as edges,

contrast, and brightness) while minimizing processing on aspects humans are less sensitive to (such as fine color details). This allows for more efficient compression, enhancement, or analysis of images while maintaining visual quality where the HVS is most responsive.

1.4.2 Why Do We Use HVS

The HVS model is employed primarily to enhance image quality and efficiency in various applications. Since the HVS is more sensitive to certain visual elements like edges and brightness, and less sensitive to others like fine color details, algorithms can focus processing power on the most important aspects, reducing the need for resources like bandwidth or storage in fields like image compression and transmission. For instance, it informs techniques where certain high-frequency details that are less perceptible to the human eye can be discarded without significant loss of perceived quality.

This approach also helps in designing displays and visual media that cater to human perception, ensuring that images are rendered in a way that aligns with how we naturally see(looks natural and realistic to human viewers) and interpret visuals for applications like Computer graphics, VR or Video Games.

HVS-based models enhance the parts of an image that are most important for human perception. For instance, in image enhancement techniques or display technologies, focusing on contrast or sharpness makes images clearer and more appealing to the human eye. This what made HVS is one of the best methods for Biometric Recognition Systems like Iris Recognition.

1.4.3 Historical Approach of HVS

The human visual system (HVS) is one of the oldest and most well studied systems, allowing researchers from many disciplines to develop image processing techniques. The journey begins from biological studies which slowly moves into the ambitious world of modern image processing applications.

1.4.3.1 Early Biological Studies

The fundamentals of HVS research date as far back to the 19th century, when researchers began revealing themysteries behind human vision. Hellmholtz to the rescueIn 1866 Hermann von Helmholtz proposed what is known as the trichromatic theory where honest-to-goodness color receptors on our retinas are sensitive to red, green and blue light. It became a basis theory to understand perception of color.

In the early 20th century, Ewald Hering introduced the opponent process theory, which proposed that color vision is based on three opponent channels: red-green, blue-yellow, and black-white. These theories helped explain various color vision phenomena and laid the foundation for color image processing.

1.4.3.2 Neurophysiological Discoveries

The advent of electrophysiology in the mid-20th century enabled researchers to study the responses of single neurons in the visual system. David Hubel and Torsten Wiesel employed the cat visual cortex in the 1950s while investigating early opportunities to reveal that primary visual cortex (V1) neurons respond selectively to oriented edges and bars[7]. This finding provided new insights about the processing spatial information in the visual system.

Additional experiments demonstrated the presence of distinct pathways carrying visual information, including magnocellular (M) and parvocellular (P) channels that are sensitive to different aspects of vision. The M pathway, which is more sensitive to low spatial frequencies and motion, is thought of as color blind whereas the P pathway, with its greater sensitivity to high spatial frequencies and color perception,was considered to be used primarily for form perception. This fact has been considered in image and video compression methods.

1.4.3.3 Psychophysical Experiments

Psychophysical experiments have played a crucial role in understanding the capabilities and limitations of human vision. In the 1960s, Campbell and Robson conducted experiments using sinusoidal gratings to measure the contrast sensitivity function (CSF) of the human visual system[8]. The CSF describes the sensitivity of the HVS to different spatial frequencies and has been widely used in image quality assessment and compression algorithms.

Other psychophysical experiments have revealed various visual phenomena, such as the Mach band effect, which demonstrates the HVS's sensitivity to changes in luminance, and visual masking, where the presence of one stimulus affects the perception of another[9]. These findings have been incorporated into image processing techniques to improve perceptual quality and efficiency.

1.4.3.4 Computational Models

As our understanding of the HVS grew, researchers began to develop computational models to simulate its behavior. In the 1980s, Daugman introduced the use of 2D Gabor filters to model the receptive fields of neurons in the primary visual cortex. Gabor filters have since been widely used in image processing tasks such as texture analysis and iris recognition.

Other computational models, such as the Contrast Pyramid[10] and the Steerable Pyramid, have been developed to mimic the multi-scale and multi-orientation processing capabilities of the HVS. These models have been applied in areas like image compression, denoising, and enhancement.

1.4.4 Algorithms Based on HVS

1.4.4.1 Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transform is a mathematical technique used to decompose an image into different frequency components at various scales. It provides a multi-resolution analysis of the image, allowing for both spatial and frequency domain representation[11].

DWT mimics how humans perceive visual information at different scales. The HVS is sensitive to certain spatial frequencies, particularly low frequencies that represent coarse details, and less sensitive to high frequencies that capture fine details. By focusing on these perceptually relevant frequencies, DWT can effectively compress images while preserving important visual features.

1.4.4.2 Gabor Filters

Gabor filters are linear filters used for texture analysis and feature extraction in images. They are designed to capture specific orientations and frequencies, making them effective for edge detection and texture representation[12].

Gabor filters are inspired by the receptive fields of neurons in the human visual cortex, which respond selectively to specific orientations and spatial frequencies. This correspondence allows Gabor filters to model human perception of texture and edges, making them useful in applications like facial recognition and iris recognition.

1.4.4.3 Contrast Sensitivity Function (CSF)

The Contrast Sensitivity Function quantifies the ability of the human visual system to detect contrast at various spatial frequencies. It is often used as a band-pass filter in image processing to assess image quality or to enhance images based on human sensitivity[13].

CSF directly reflects how humans perceive contrast and detail in images. By applying CSF as a filter during image processing, algorithms can prioritize features that are more likely to be noticed by viewers, thus improving perceptual quality.

1.4.4.4 Chroma Subsampling

Chroma subsampling is a technique used in image compression where color information is sampled at a lower resolution than brightness information. This method reduces data size while maintaining perceived image quality.

This technique exploits the fact that human vision is more sensitive to luminance (brightness) than chrominance (color). By reducing color resolution without significantly affecting perceived quality, chroma subsampling aligns with how humans naturally process visual information[14].

1.5 Hardware Implementation

Iris recognition is a biometric technology that identifies individuals by analyzing unique patterns in the iris of the eye. The process typically involves several stages: image acquisition, pre-processing, feature extraction, and matching. These operations often require substantial computation, especially in real-time applications, such as security systems, border control, or mobile devices.

To meet the demand for speed, accuracy, and efficiency, iris recognition systems can benefit from hardware implementation. Implementing iris recognition on hardware platforms like FPGA

enables faster processing, parallel execution, and the ability to deploy systems in low-power real-time environments. In addition, these hardware systems can be embedded in small, portable devices, offering a compact solution for systems like mobile biometric scanners or smart cameras that include iris recognition as part of a broader system.

1.5.1 Different Hardware Implementations

- **FPGA :** FPGAs are integrated circuits that can be configured by the user after manufacturing. They consist of an array of programmable logic blocks and interconnects, allowing for custom hardware implementations of algorithms. It provides Parallel Processing, Customization and Low Latency. FPGA are used where space, power consumption, and processing speed are critical factors. They can be found in standalone iris recognition devices or integrated into larger biometric identification systems.

- **Application-Specific Integrated Circuits (ASICs) :** ASICs are custom-designed chips tailored for a specific application or function. Unlike FPGAs, which can be reprogrammed, ASICs are fixed once manufactured. ASICs are commonly used in commercial iris recognition products where high volume and efficiency are required. They may be employed in security systems, mobile devices, and biometric terminals.

- **Digital Signal Processors (DSPs) :** DSPs are specialized microprocessors designed specifically for digital signal processing tasks. They excel at performing mathematical operations on signals quickly and efficiently.

- **General-Purpose Processors (CPUs) :** While not as efficient as FPGAs or ASICs for specific tasks, CPUs may still be used in iris recognition systems for prototyping or less demanding applications where real-time performance is not critical.

1.6 Conclusion

this chapter lays the foundational framework necessary to understand the technical and theoretical aspects of iris recognition systems within the broader context of biometric recognition. It highlights the significance of iris recognition by discussing the unique challenges and advantages associated with it, while also connecting these concepts to the Human Visual System, which plays a critical role in both biological and machine-based recognition processes. Furthermore, the chapter introduces key hardware platforms used for the implementation of recognition systems, with a focus on FPGAs as the chosen platform for this thesis. The insights and knowledge provided here will serve as a basis for the more detailed technical discussions that will follow in subsequent chapters.

Chapter 2

State of the Art Methods

2.1 Introduction

This chapter presents a review of the state-of-the-art methods and approaches related to Human Visual System (HVS) algorithms and iris recognition systems. We begin by exploring various HVS-based algorithms documented in the literature, along with their key applications in biometric systems and image processing. Following this, the chapter delves into the current advancements in iris recognition, focusing on the different techniques used for preprocessing, iris detection, segmentation, normalization, feature extraction, and classification. Finally, we review the hardware platforms and programming environments commonly employed for implementing these systems.

2.2 Human Visual System State of the Art Models and Their Applications

In this section, we will explore various state-of-the-art methods and algorithms that leverage principles of the Human Visual System (HVS) as highlighted in the literature. By examining the core ideas behind these methods, we aim to illustrate how they capitalize on the unique characteristics of human visual perception. Each method will be presented alongside its practical applications, demonstrating the importance of aligning technological advances with biological insights to enhance image processing, recognition, and analysis. This comprehensive review underscores the critical role of the HVS in informing modern approaches to visual tasks, ultimately advancing our understanding and capabilities within the field.

- In their 2018 study, *Gursoy et al.*[15] proposed the use of Discrete Wavelet Transform (DWT) as a robust feature extraction technique for classifying power quality disturbances. The authors highlighted the ability of DWT to decompose electrical signals into different frequency components, which allows for effective identification of disturbances such as voltage sags, swells, and transients in power systems. By applying DWT to real-time data, they demonstrated improved accuracy in classifying various power quality events compared to traditional methods. Their findings emphasized that DWT not only enhances the classification process but also contributes to better monitoring and management of electrical power systems, making it a practical tool for utilities facing increasing demands on power quality. The figure 2.1 shows the decomposition made by the DWT on an image:

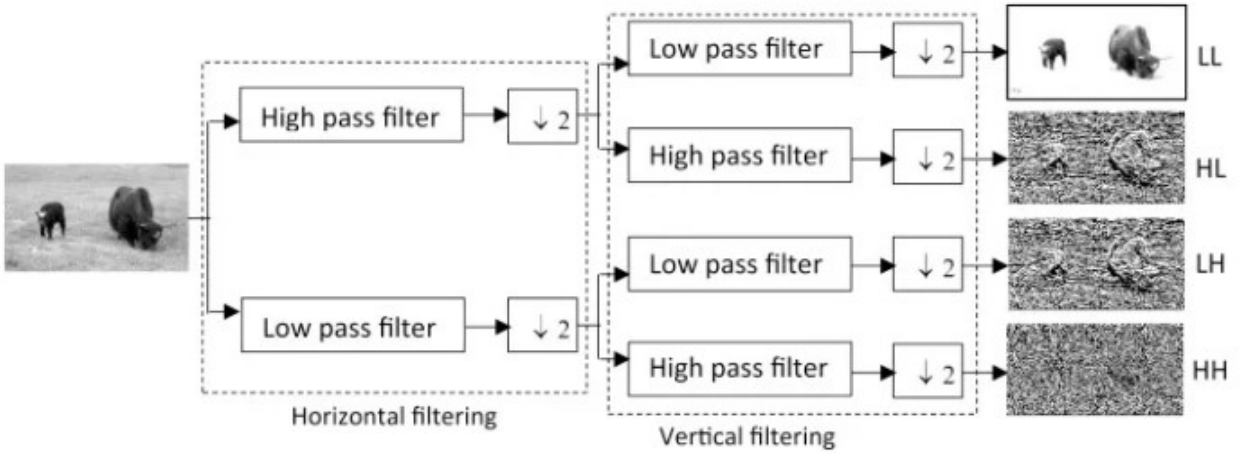


Figure 2.1: Discrete Wavelete Transform

- In their seminal work, *Daly et Al*[16] introduced the Visible Differences Predictor (VDP) algorithm, which predicts perceived image quality by modeling human visual characteristics. The authors emphasized that traditional metrics often fail to align with human perception, leading to misleading assessments of image quality. The VDP incorporates factors such as contrast sensitivity and visual masking, allowing for a more accurate prediction of how differences between images will be perceived by viewers. Their algorithm is particularly beneficial in image compression and enhancement applications, ensuring that visual fidelity aligns closely with human perception. This work significantly influences the fields of image processing and computer graphics by providing a reliable metric for assessing image quality. The figure 2.2 shows the general architecture of the Visible Differences Predictor suggested by *Daly*[16] :

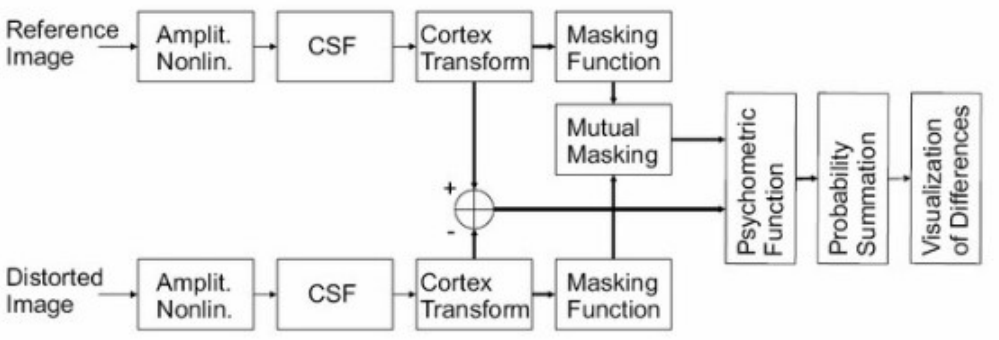


Figure 2.2: Visible Differences Predictor

- *Daugman et al*[17] made significant contributions to the field of image processing with their discussion on the use of Gabor filters for texture analysis and feature extraction. This seminal paper emphasized the biological relevance of Gabor filters to the human visual system (HVS), as they mimic the receptive fields of visual neurons. The authors demonstrated that Gabor filters are effective for capturing spatial frequency and orientation information in images, which are crucial for texture recognition. By employing these

filters, they provided insights into how the HVS interprets complex visual stimuli, paving the way for advancements in facial recognition, iris recognition, and other applications requiring detailed texture analysis.

The figure 2.3 shows the application of apllying a gabor filter bank generated by varying frequency and orientation on an image of a circle and the results of its implementation :

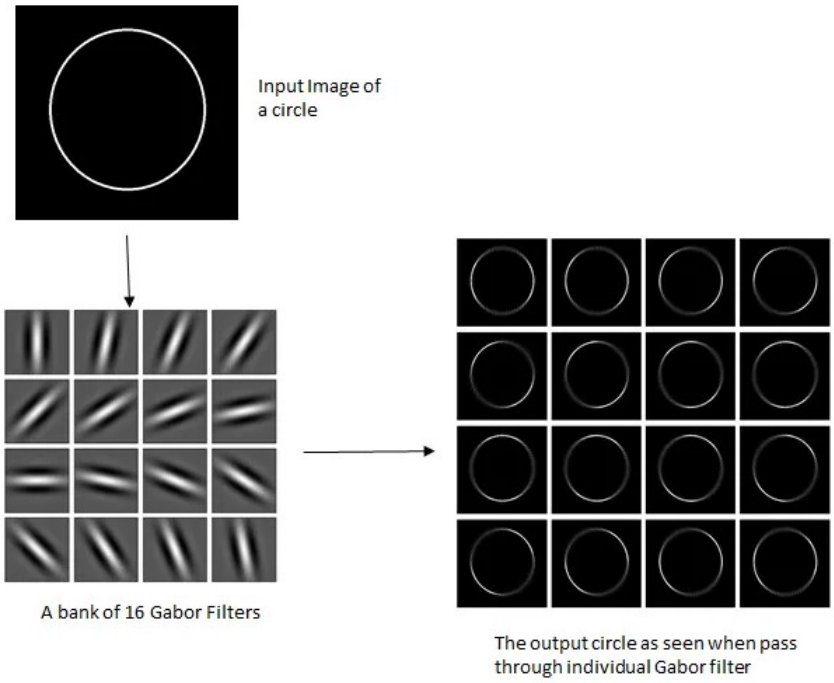


Figure 2.3: Exemple of Gabor Filter

- Other research[13] defined the Contrast Sensitivity Function (CSF) and its implications for understanding human visual sensitivity to different spatial frequencies. CSF quantifies human sensitivity to different spatial frequencies

- Another paper[18] introduced log-polar transforms as a method for analyzing orientations in images, based on principles of human visual perception. It discussed how log-polar coordinates can effectively represent images while preserving essential features related to orientation and scale. This approach mimics the way humans perceive visual information, particularly in the context of image registration and image compression.

- *Bojilov and Bocheva*[19] developed a neural network model aimed at enhancing the visual discrimination of complex motions, addressing the challenges inherent in recognizing intricate movement patterns. Their model is inspired by the Human Visual System (HVS) and employs a multi-layer neural network architecture that simulates the visual processing pathways of the human brain. Through training with diverse motion stimuli, the authors demonstrated the model's capability to accurately differentiate between various trajectories and velocities, capturing the nuances of complex motions. They highlighted potential applications in fields such as robotics, video analysis, and human-computer interaction, where precise motion perception is essential. This work underscores the significance of aligning artificial intelligence with biological insights, paving the way for advancements in visual processing technologies.

2.3 Iris Recognition System State of the Art Models

2.3.1 Image Preprocessing

The initial sub-stage of detection is preprocessing, which encompasses several crucial steps to prepare the image for further analysis. This phase begins with image acquisition, where high-quality images of the face are captured to ensure accuracy. Following this, image enhancement techniques are applied to improve the overall quality of the image. Examples of such techniques include adjusting contrast and reducing noise. These preprocessing operations are essential for eliminating unwanted artifacts and making the image more informative, thereby optimizing it for the subsequent feature extraction phase.

In this stage, recent advancements have significantly contributed to the field. For instance, Zhang et al[20]. proposed a novel image enhancement technique utilizing deep learning-based filters that adaptively enhance image quality by learning from large datasets. This approach has shown promising improvements in system accuracy.

In the work of Ming Liu[21], a Fuzzy method was employed to enhance images. Unlike other approaches, this method focused on addressing the image non-linearities that are commonly ignored by commonly used filters such as Median, Gaussian, and others. These filters typically assume the image characteristics to be Gaussian, which is often not the case.

Another interesting work by Chang et al[22]. aimed to isolate the pupil and locate the inner boundaries of the iris. They proposed a preprocessing method consisting of three stages: Gaussian filtering with a low value of sigma then binary thresholding, and a subsequent Gaussian filter. This method effectively detected the pupil and isolated it from the rest of the eye.

2.3.2 Edge Detection

After the preprocessing phase, the phase of detecting and enhancing the iris area takes place, Several researchers including *Jasem Rahman et al.*[23] highlighted the use of Sobel edge detection as a key technique in iris recognition systems. Sobel edge detection, known for calculating image intensity gradients, is used to extract critical iris features by detecting edges, such as ridges and furrows, that define an individual's unique biometric signature. The authors emphasized its efficiency in handling noise and capturing fine details, making it essential for the preprocessing stage of iris recognition.

Omran[24] Utilized the Canny edge detector as a method to highlight the boundaries of the iris, given its efficiency and effectiveness, The Canny edge detector has been adopted in iris recognition systems to detect and enhance the edges of the iris region.

In their study, *Sundas Naqeeb Khan et al.*[25] explored the application of Laplacian of Gaussian (LoG) edge detection for iris recognition, particularly in the context of flight simulator user identification. LoG edge detection combines Gaussian smoothing with the Laplacian operator to identify edges by detecting areas of rapid intensity change, making it effective in isolating important iris patterns. The authors highlighted that LoG excels in reducing noise while capturing precise iris details, which are critical for accurate identification in dynamic environments like flight simulators. Their findings demonstrated that LoG enhances the reliability of iris recognition by providing robust edge detection, leading to improved user authentication in

high-performance scenarios

2.3.3 Segmentation

Various approaches have been used for the localization of iris boundaries takes place, retaining only the iris region, as mentioned in [26], including K-means clustering, Active contour models, Edgeless active contour, Gradient vector flow snake, Statistical learning methods, least median square differential operator and linear basis function.

In their review, *Hind Hameed Rasheed et al.*[27] examined the use of the Hough Transform for iris segmentation in biometric systems. The Hough Transform is employed to accurately detect circular shapes, making it particularly effective in identifying the iris boundary within an eye image. By transforming edge-detected pixels into a parameter space, the algorithm identifies the iris's circular region, isolating it from surrounding features like the pupil and sclera. The authors highlighted that the precision of the Hough Transform in segmenting the iris plays a crucial role in improving the accuracy of subsequent recognition stages, especially in challenging images. This method is essential for reliable and efficient iris recognition, which is critical for enhancing biometric applications.

Lian et al.[28] proposed the Attention U-Net (ATT-UNet) for iris image segmentation, building on the standard U-Net architecture by incorporating an attention mask. This mask helps the model learn more distinctive features, improving the classification of iris and non-iris pixels. The process begins by selecting a bounding box for the potential iris region, followed by the generation of attention masks, which are used as weighted functions to refine feature maps. ATT-UNet has a structure similar to VGG16 but without fully connected layers, and its weights are initialized using a pretrained ImageNet model. Despite its effectiveness, the method faces challenges such as occlusion, reflections, poor lighting, and blurring. To address these, *Zhang et al.*[29] proposed four network schemes combining dilated convolution with U-Net to enhance iris segmentation performance. The best-performing model, FD-UNet, fully integrates dilated convolutions, improving both the efficiency and accuracy of segmentation.

2.3.4 Normalization

Normalization is a crucial step in iris recognition, transforming the circular iris region into a rectangular format to enable easier processing. A widely used approach for this task is Daugman's Rubber Sheet model[30], which, though introduced years ago, remains highly efficient and continues to be a key method in modern research due to its effectiveness.

Figure 2.4 shows the principle of transforming the circular iris into a rectangular shape.

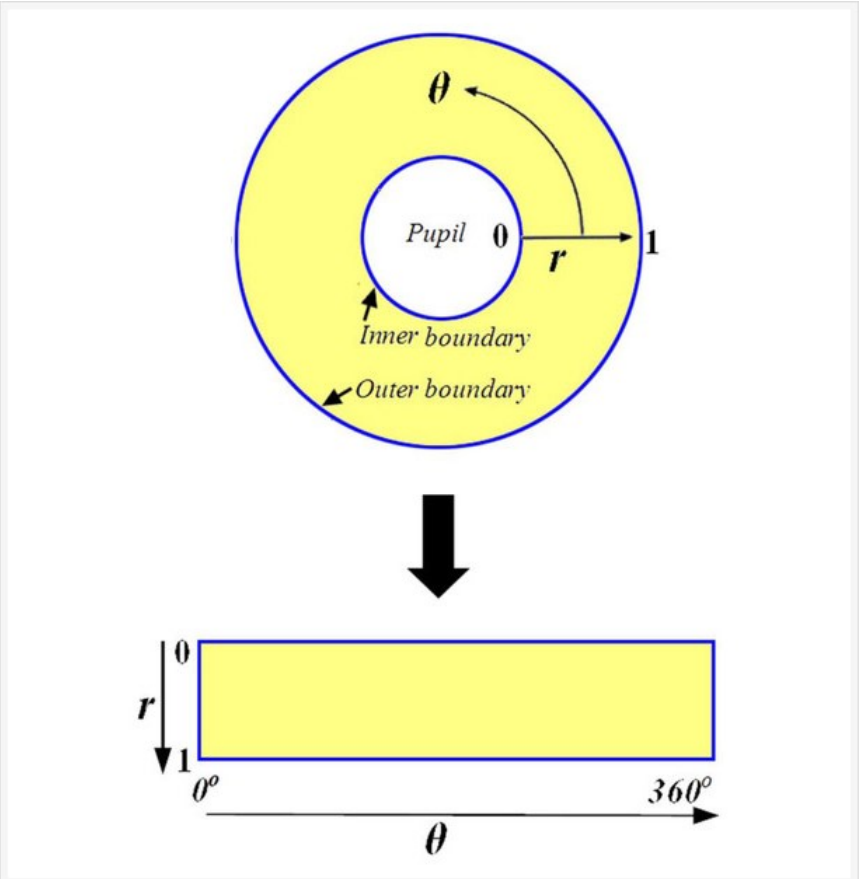


Figure 2.4: Daugman's Rubber Sheet Normalization

2.3.5 Feature Extraction

To extract the most important features of the iris after the segmentation and normalisation phase, various mathematical algorithms have been implemented for dimension reduction, including Principal Component Analysis based on Discrete Wavelet, Haar and Biorthogonal Wavelet Transform, Gabor and Log Gabor filters, Local Binary Patterns, Discrete Cosine Transform, Gray Level Co-occurrence Matrix [31], Residual Pooling layer introduced in the Convolutional Neural Networks [32], Joint Bayesian Formulation and Supervised Discrete Hashing [33], .

2.3.6 Classification

After extracting the most significant features, the final stage of the system is classification. In iris recognition, various classification approaches are employed, ranging from traditional methods to more advanced machine learning and deep learning techniques.

Some articles like[34] used The Hamming Distance which is a classical method. The Hamming Distance is ideal for iris recognition due to its speed and efficiency, allowing for quick comparisons between binary codes while being resilient to noise and small variations in the iris image, such as those caused by lighting conditions or occlusions.

Others used Machine Learning and Deep learning Based methods, for deep learning methods we can Convolutional Neural Networks, as seen in [21], Machine Learning based methods used

are K-Fold Cross-Validation, K-Nearest Neighbor [35] and Support Vector Machine, utilized in, [31] and [32],

In addition, a hybrid approach is mentioned in [36], where both the Hamming distance and a neural network are used for the matching process.

2.4 Databases

Iris images are typically captured using either the visible wavelength spectrum (VIS) or the near-infrared (NIR) spectrum, with four key components involved in the acquisition process: lighting, lens, sensor, and console. NIR-based images focus on the complex iris texture rather than pigmentation, making them less prone to noise and effective even for dark-colored irises, thus improving recognition performance. These images are captured in both ideal (constrained) and non-ideal (unconstrained) environments. Table 2.1 provide past research works with their respective testing databases.

Source	**Database**	**Subjects**	**Images**	**Spectrum**	**Resolution**	**Format**
Drozdowski et al.[37]	CASIA-1000	1,000	20,000	NIR	640×480	.jpeg
De Marsico et al.[38]	CASIA.V1	108	756	NIR	320×280	.bmp
Zhang et al.[39]	CASIA.V3	1614	22,548	NIR	640×480	.jpeg
Arsalan et al.[40]	CASIA.V4	3284	32,537	NIR	640×480	.jpeg
Proenca et al.[41]	UBIRIS.V2	261	11,102	VIS	800×600	.tiff
Kim et al. [42]	BERC mobile	75	3,011	NIR	960×1280	—
Proença et al.[43]	UBIRIS.V1	241	1,249	VIS	800×600	.jpeg
Dobes et al.[44]	UPOL	64	384	VIS	786×576	.png
Rattani et al.[45]	UBIPr periocular	344	5,126	VIS	—	.bmp
Khan et al.[46]	ND-IRIS-0405	356	64,980	NIR	640×480	.jpg
Wang et al.[47]	PolyU bispectral	209	12,540	NIR, VIS	640×480	—
Cheng et al. [48]	JLU-4.0 Iris	88	26,400	NIR	640×480	.bmp
Rakshit [49]	Bath800	800	31,997	NIR	1280×960	.j2c
Umer et al.[50]	IITD	224	2,240	NIR	320×240	.bmp
Sequeira et al. [51]	MobBIO	105	1,680	VIS	300×200	.jpg
Arsalan et al. [52]	NICE-II	67	1,000	VIS	400×300	.tiff
Chen et al.	VSIA [53]	55	550	VIS	—	—

Chirchi et al. [54]	MMU	100	995	NIR	320×240	.bmp
Fierrez et al. [55]	BioSec-baseline	200	3,200	NIR	—	—
Wang et al. [56]	Cross-Eyed-2016	120	3,840	NIR, VIS	400×300	.jpg
Fenker and Bowyer [57]	ND-Iris Template-Aging	200	11,776	NIR	—	.tiff
Lucio et al. [57]	CSIP	50	2,004	VIS	—	—

Table 2.1: Overview of iris databases, number of subjects, images, spectrum, resolution, and file formats

2.5 Discussion

Table 2.2[23] shows the performance efficiency rate of various traditional and deep learning techniques in some previous works of the Iris Recognition System with the metrics being used are Accuracy rate (ACC) or error rate (EER).

Source	**Database**	**Feature extraction methods**	**Classification methods**	**ACC/EER (%)**
Liu et al.[58]	CASIA, Q-FIRE	Pairwise filter bank	CNN	EER = 0.31
Kumar et al.[59]	IITD, MMU	DWT and DCT	ED	94.59
Salve and Narote[60]	CASIA-IrisV4	1D log-Gabor wavelet	ANN and SVM	92.5–95.9
Du et al.	NIST-ICE[61]	Flipped images	They used the CNN technique	97.5–98.6
Alvarez-Betancour and Garcia-Silvente[62]	CASIA, UBIRIS, MMU	Fusion of three information sources of SIFT features	The nearest neighbor classifier	99.90, 99.89, 99.05
Al-Waisy et al.[63]	SDUMLA-HMT, CASIA-V3, IITD	CNN	Softmax classifier + fusion	100, 99.92, 99.88
Sahu et al.[64]	CASIA-V3, BATH	PILP and DBSCAN	Clustering approach	97.68, 96.23
Gad et al.[65]	SDUMLA-HMT, CASIA v.1, UBIRIS v.1, CASIA v.4	Delta mean (DM) and multi-algorithm mean (MAM)	Euclidean distance (ED)	99.49, 99.46, 99.46, 99.49
Barpanda et al.[66]	CASIA, UBIRIS, IITD	Tunable filter bank	Canberra distance	91.65

Liu et al.[67]	MICHE-I: Galaxy S4, iPhone 5	Proposed multilayer analogous convolutional architecture inspired by CNN	Collaborative representation scheme	97, 98
Wang and Kumar [68]	ND-IRIS-0405, CASIA-distance, WVU non-ideal	Residual learning and dilated kernel utilizing deep CNN	The Hamming distances	97.7, 87.5, 96.1
Barpanda et al.[69]	CASIA, IITD, UBIRIS	Wavelet derived from biorthogonal Cohen–Daubechies–Feauveau	Canberra distance	91.65
Zhao and Kumar[?]	ND-IRIS-0405-Iris, CASIA-distance, WVU non-ideal	FCN-based framework for spatially corresponding descriptors of iris features	Hamming distance (HD) from extended masks and binarized feature maps	97.1, 84.1, 94.3
Wang and Kumar [70]	PolyU bi-spectral, Cross-eyed-cross-spectral iris	CNN similar to AlexNet architecture	SDH for compression and classification	90.71, 87.18
Reddy et al.[71]	VISOB, UBIRIS-I, UBIRIS-II, Cross-eyed	OcularNet based on CNN architecture; trained PatchCNN for every patch	ED technique between patch's embedded iris feature	EER=1.89, EER=9.86, EER=9.77, EER=14.9
Zhao and Kumar[72]	ND-IRIS-0405, CASIA-distance, IITD iris, WVU non-ideal	FeatNet model for extracting features of iris region based on FCN	Fractional Hamming distance (HD) for extended masks and binarized feature maps	EER = 1.12, EER=4.07, EER=0.76, EER=2.20
Lee et al.[73]	CASIA-Distance, CASIA-Lamp, CASIA-1000	Deep ResNet-based recognition	ED to classify CNN's fully connected layer output	EER=2.16, EER=1.59, EER=1.33
Ahmadi et al.[74]	CASIA-V3, UBIRIS-V1, UCI datasets	Hybrid 2-DGK/SF/PF techniques for iris texture features	Intelligent hybrid RBFNN-GA classifier	99.9914, 99.9889, 99.98
Lee et al.[75]	NICE.II, MICHE, CASIA-distance	Three CNN models for feature extraction	SVM as classifier	EER=8.58, EER=17.39, EER=2.96

Table 2.2: Comparison of various iris recognition methods across different databases, feature extraction, and classification techniques.

Many methods achieve near-perfect accuracy, with values reaching up to 100% (Al-Waisy et al. and Ahmadi et al.), indicating that current iris recognition systems are highly reliable under

controlled conditions.

EER (Equal Error Rate) values, particularly for databases like CASIA and UBIRIS, show that CNN-based methods consistently reduce error rates. The lowest EER values are achieved by models such as OcularNet (Reddy et al.) and FeatNet (Zhao and Kumar), which integrate deep learning and fine-tuned classification techniques.

Comparatively, older methods like Canberra distance-based classifiers or traditional filter bank methods still perform well.

2.6 Hardware Implementations

2.6.1 Writing directly with VHDL

The most common approach involves using HDL directly for coding pre-trained deep learning applications, with training performed off-chip. *Hossam et al.*[76] present a design methodology for implementing a Multiply-Accumulate (MAC) unit using VHDL, tailored to support deep learning networks on FPGA platforms. The authors emphasize the concurrent nature of the MAC unit, which enables it to perform multiple multiply-accumulate operations simultaneously, enhancing the efficiency of deep learning computations.

2.6.2 Using Matlab

Another notable recent approach is the use of Matlab. The Deep Learning Toolbox [77] and the Deep Learning HDL Toolbox [78], both part of the MATLAB environment, assist in designing and deploying CNN models on FPGA platforms.

Implementing a CNN model on an FPGA (based on their official documentation [79]) follows the steps :

1. Import the pre-trained model
2. Perform Fixed-Point Conversion,
3. Generate the HDL code,
4. Generate the IP core ,
5. Create the bitstream file,
6. Deploy the model on the FPGA.

2.6.3 Using high level Synthesis Method

Bing Liu et al. utilized the HLS (High-Level Synthesis) technique [80]. HLS is a process that converts high-level code into Hardware Description Language (HDL), enabling programming of FPGAs. The approach follows these steps:

- Initially, the functions intended for deployment on the FPGA are written in C or C++.
- These functions are then synthesized into VHDL (or Verilog) to generate an IP block specifically designed for the required functionality.
- Finally, the complete system is created by building a block design that integrates the generated IP blocks. This block design is then synthesized and implemented on the FPGA to complete the process.

2.6.4 Using AI tools

FPGA vendors like Xilinx and Intel provide complete toolchains for AI model development and deployment on FPGAs:

- **Xilinx Vitis AI :** A comprehensive AI inference development platform, which includes optimized models, compilers, and tools for deploying on FPGA or ACAP (Adaptive Compute Acceleration Platform). *Wang and al.*[81] discussed the implementation of an Object Detection Accelerator on an FPGA using the Vitis-AI development framework.
- **Intel OpenVINO :** Primarily designed for computer vision AI applications, it includes FPGA acceleration as part of its suite, converting AI models into formats compatible with FPGA hardware.

So to summarize the first four methods, to implement our method in an fpga we use the following steps :

1. **Model Training:** AI models are typically trained on GPUs or CPUs using frameworks like TensorFlow, PyTorch, or MATLAB.
2. **Optimization:** AI tools perform optimizations like quantization, pruning, or fixed-point conversion to make the models suitable for FPGA hardware.
3. **HLS or Compilation:** High-Level Synthesis (HLS) tools convert the AI model into HDL code (e.g., Verilog or VHDL), while AI-specific frameworks compile models directly for FPGA execution.
4. **Deployment:** The optimized model is implemented on the FPGA, where inference tasks are executed, delivering high performance and power efficiency.

2.6.5 Using Python for Pynq

One of the key advantages of the Pynq FPGA Cards is the ability to program the FPGA using Python, through the PYNQ framework. This is especially useful for developers who prefer working in high-level languages and want to take advantage of FPGA acceleration without writing low-level hardware description code (VHDL or Verilog).

Pynq comes with support for Jupyter Notebooks, allowing users to write Python code to control and interact with the FPGA directly from a web interface. This includes the ability to load pre-compiled FPGA bitstreams and run hardware-accelerated functions.

PYNQ also provides pre-built hardware "overlays" (FPGA bitstreams) that implement common hardware blocks, which can be controlled from Python code. These overlays are analogous to software libraries but for hardware.T hese overlays are already optimized for FPGA use and allow users to accelerate applications like image processing, matrix operations, or inference tasks.

2.7 Conclusion

In conclusion, this chapter has provided a thorough review of the HVS-based algorithms and iris recognition techniques currently utilized in the field. By examining the methods and tools used at various stages of the iris recognition process, along with the hardware platforms supporting these implementations, we have established a solid understanding of the state-of-the-art approaches. This review lays the groundwork for the development and implementation of our proposed system, which will build upon these existing techniques.

Chapter 3

Human Visual System's Terminology and Methods

3.1 Introduction

This chapter presents a synthesis of research on the analysis of the human visual system. Its goal is to provide the essential knowledge necessary for developing "bio-inspired" image analysis algorithms. Each processing step in the human visual system is an optimized process passed down through generations, enabling individuals to efficiently process visual data to adapt to their environment. Hence, we can assume that the human visual system is, for our needs, the "ideal" system capable of adapting to common situations. The human visual system serves as a reference to computer vision systems, aiming to reach a similar level of performance.

This chapter also provides the key concepts in Image Processing, Machine/Deep Learning associated with Iris Recognition Systems, This will enhance the comprehension of our chosed approach. Finally we will provide an overview of the FPGA implementation.

3.2 Anatomical and Physiological Foundations of the Human Visual System

3.2.1 Introduction

This section describes the knowledge gained about the human visual system. The presentation follows the path of visual information from photoreceptors to the brain. Understanding the human body and its functioning has always fascinated researchers, with the aim of uncovering the architecture and interactions between its various components. The visual system is no exception, and its exploration has led to the discovery of the cells and neurons that compose it, though much remains unknown about its functioning. Nevertheless, these insights offer a better understanding of the system as a whole and help in addressing certain pathologies.

Understanding the human body is not only about treating its disorders. As discoveries progress, we uncover the remarkable efficiency and ingenuity of the human body, especially in vision. Our ability to observe and comprehend our environment, near or far, familiar or unfamiliar, in everyday conditions, demonstrates that the human visual system far surpasses all current

computer vision algorithms.

As shown in this section, the visual system optimizes incoming visual data from the outset to simplify subsequent processing. Each stage has a specific function to extract particular data, benefiting from previous processing. This strategy often contrasts with approaches that introduce complex post-processing to compensate for initial data flaws.

3.2.2 General Presentation of the Human Visual System

Visual information is captured by the eye and undergoes initial processing in the retina before being transmitted to the brain. Higher-level processing occurs in the visual areas located in the back of both cerebral hemispheres as presented in Figure 3.1. Physiological studies[82] indicate that a significant portion of the cortex is dedicated to the processing of visual information. Furthermore, the intricate connections between these visual areas and other regions of the brain contribute to the complexity of the visual system. Figure 3.1 outlines the key processes examined in this study: retinal processing, followed by the transmission of information through the optic nerve and its subsequent processing in the primary visual cortex (area V1).

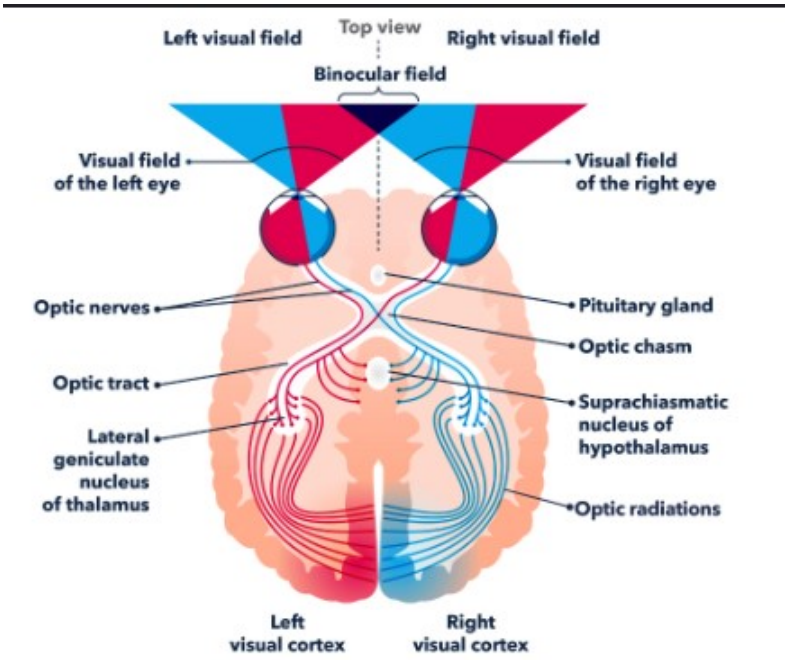


Figure 3.1: Overall diagram of the human visual system

3.2.3 The Eye

The eye, depicted in figure 3.2, functions as a receiver of light signals. Focuses these signals to create an image that is then sent to the brain through the optic nerve. The lens, though not perfect, acts as a variable-focus lens that projects the image of the observed object onto the retina, a photosensitive layer at the back of the eye.

The pupil, regulated by the iris, operates as an automatic aperture, adjusting its size based on the amount of light received.

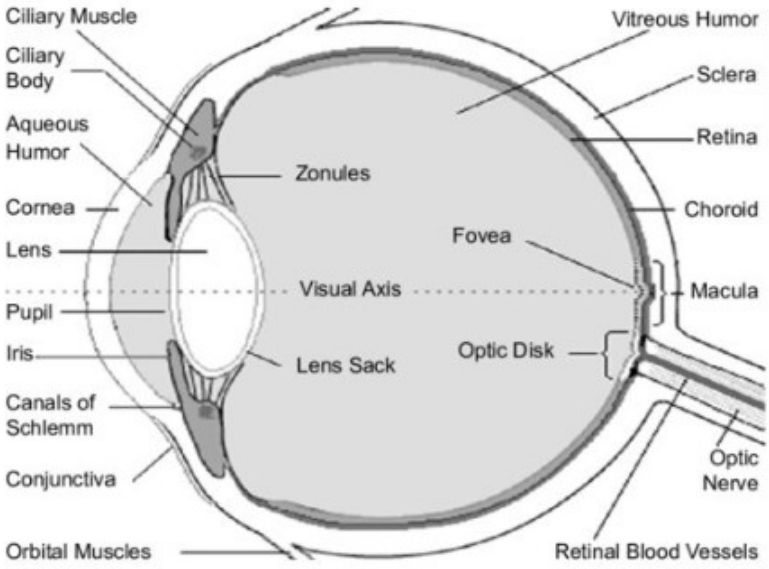


Figure 3.2: Schematic cross-section of the eye[83]

3.2.3.1 The Retina

The retina is the neurosensory layer of the eye, covering about 12.5 cm^2 [84]. Its role is to convert light into electrical signals, which are then transmitted to the brain via the optic nerve. The retina displays significant regional variations that influence its function. The central region is specialized for detailed and color vision (photopic vision), while the peripheral areas are responsible for achromatic (scotopic) and broader vision.

The retina contains several types of nerve cells: direct pathway cells (photoreceptors, cones and rods, bipolar cells, and ganglion cells), as well as indirect pathway cells (horizontal cells and three types of amacrine cells), along with pigment cells as shown in figure 3.3.

Notably, light must pass through all the retinal layers to reach the photoreceptors, except at the fovea. The fovea, located at the optical center of the eye (also known as the macula), contains only cones. In this area, each cone is connected to a single bipolar cell and a single ganglion cell, resulting in the highest visual acuity.

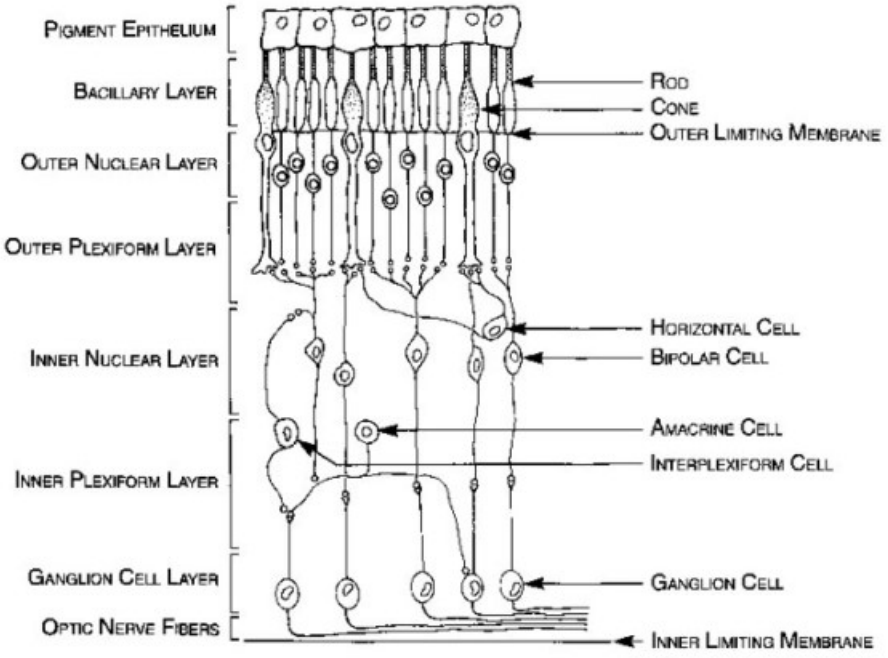


Figure 3.3: Organization of the retinal cell layers[85]

At the optical center, the fovea creates a small depression where light directly falls on the cones. The optic disc, the starting point of the optic nerve, is situated on the nasal side, approximately 4 mm away from the fovea. This region lacks photoreceptors, making it insensitive to light, and is known as the blind spot.

3.2.3.2 The fovea

The fovea is positioned directly along the eye's visual axis, where the retina is extremely thin because it contains only cones, with other visual cells pushed to the periphery. This region has about 2,500 cones, each connected to a single associative cell. Consequently, it provides the highest visual acuity. As you move away from the fovea, visual acuity decreases rapidly, making it necessary to align the eye with the object of interest for clear vision. This alignment is controlled by the extra-ocular muscles attached to the eye's capsule. The visual field extends up to 95 degrees on the temporal side from the point of fixation, but only 60 degrees on the nasal side, 50 degrees upward, and 80 degrees downward.

3.2.3.3 Photoreceptors

Photoreceptors in the human retina, namely rods and cones, have specialized roles that are fundamental to our visual perception.

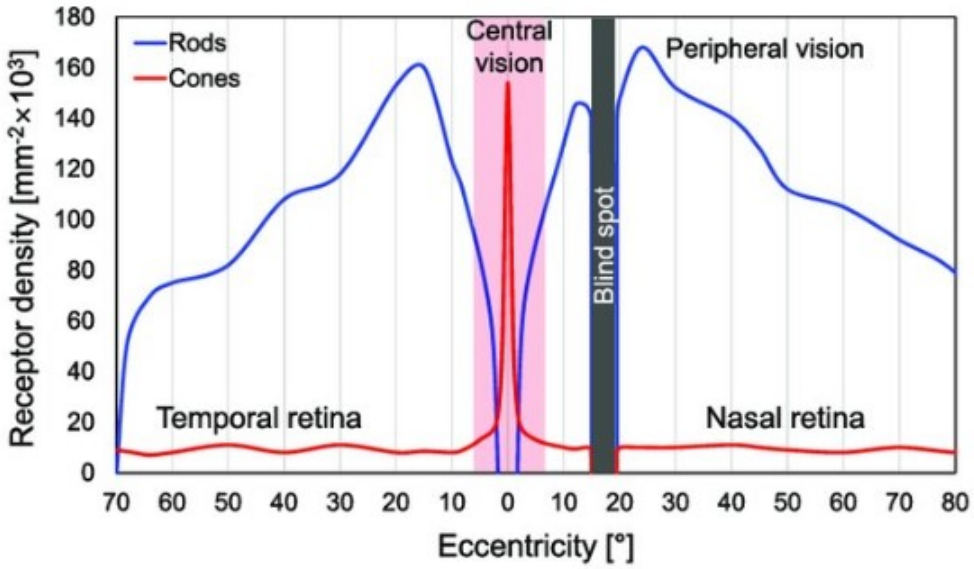


Figure 3.4: Distribution of rods and cones depending on the eccentricity from the fovea of the human eye[86]

- **Rods :** with a numerous number estimates ranging from 75 to 150 million, are predominantly located in the peripheral regions of the retina, away from the fovea. These photoreceptors are relatively large, averaging 2.5 to 3 microns in diameter, and are spaced approximately 10 to 20 microns apart. This distribution allows rods to cover a broad area, but results in lower spatial resolution compared to cones. Despite their lower resolution, rods are highly sensitive to low light conditions, making them essential for scotopic vision, which is crucial for night-time or low-light environments. Rods possess high sensitivity to light but lack the ability to distinguish colors, as they are not responsive to different wavelengths of light.
- **Cones :** Cones in contrast to Rods, are fewer in number, with approximately 6 to 7 million distributed mainly within the fovea, the central region of the retina. The cones are smaller, with diameters ranging from 1 to 2 microns, and are packed more densely, with distances between them ranging from 2.5 to 10 microns. This dense arrangement in the fovea is critical for high spatial resolution and is vital for photopic vision, which occurs under bright lighting conditions. The cones are responsible for color vision and visual acuity during the day. There are three types of cones, each sensitive to different wavelengths of light: long-wavelength (L) cones, which respond to red light; medium-wavelength (M) cones, which respond to green light; and short-wavelength (S) cones, which are sensitive to blue light. This trichromatic system allows for the perception of a wide range of colors and fine details in well-lit environments.

The spatial distribution and sensitivity of rods and cones also play a role in the adaptive mechanisms of the visual system. Photoreceptors adjust their dynamic range to match the ambient luminance through a process known as adaptive compression. This mechanism allows the visual system to handle a wide range of light intensities, although individual photoreceptors can only respond effectively over a limited dynamic range of about one to two log units. This adaptive response is illustrated by the varying sensitivity of photoreceptors to different levels of luminance, as shown in the figure 3.5.

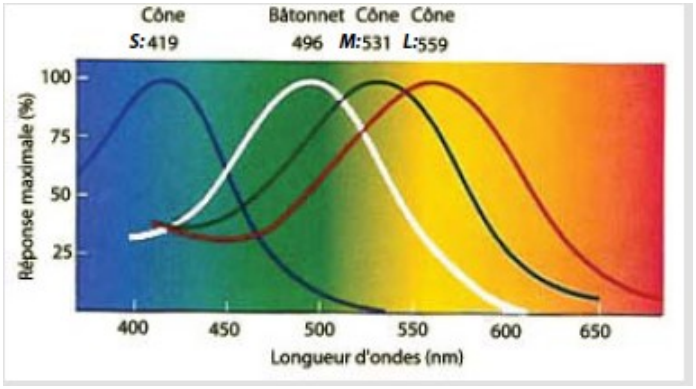


Figure 3.5: The adaptation of photoreceptor sensitivity to varying light levels[87]

3.2.4 The Optical Nerves and the Brain

3.2.4.1 Visual and Optical Nerves and Chiasm

The information gathered by the retina is transmitted via the optic nerve, which relays visual data to the brain. This process involves several key structures. First, the optic nerve carries the visual signals from each eye to the optic chiasm, where the nerve fibers from the nasal halves of each retina cross to the opposite side of the brain. This crossing ensures that visual information from the right visual field is processed by the left hemisphere and vice versa.

From the optic chiasm, the visual signals travel through the optic tract to the lateral geniculate nucleus (LGN) of the thalamus. The LGN acts as a relay station, where the signals are further processed and refined. There are six layers within the LGN, each receiving input from different types of retinal ganglion cells and contributing to various aspects of visual processing, such as color and motion.

After processing in the LGN, visual information is sent to the primary visual cortex (V1) in the occipital lobe of the brain. Here, more complex processing occurs, including the integration of spatial and temporal aspects of the visual scene. The visual cortex is responsible for higher-level processing, including depth perception, object recognition, and visual consciousness. This intricate pathway ensures that visual information is accurately interpreted, allowing coherent and meaningful visual experiences.

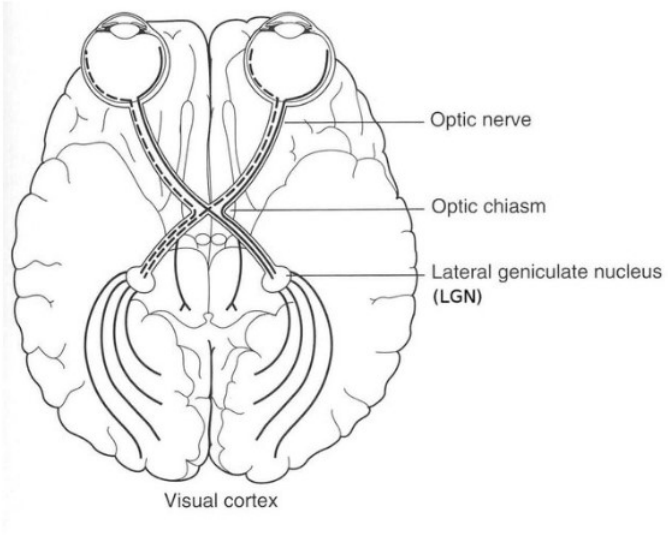


Figure 3.6: Schematic view of the visual pathway[88]

3.2.4.2 The Visual Cortex

The visual cortex is the final step in the visual perception process before the brain processes information. It is responsible for thought, language, and will. It analyzes sensations and triggers voluntary movements. This layer of neurons is responsible for all the higher functions of the organism.

The visual cortex is divided into two parts:

- The left hemisphere gathers, processes information, and controls movements on the right side of the body.
- The right hemisphere performs the same function, but for the left side of the body.

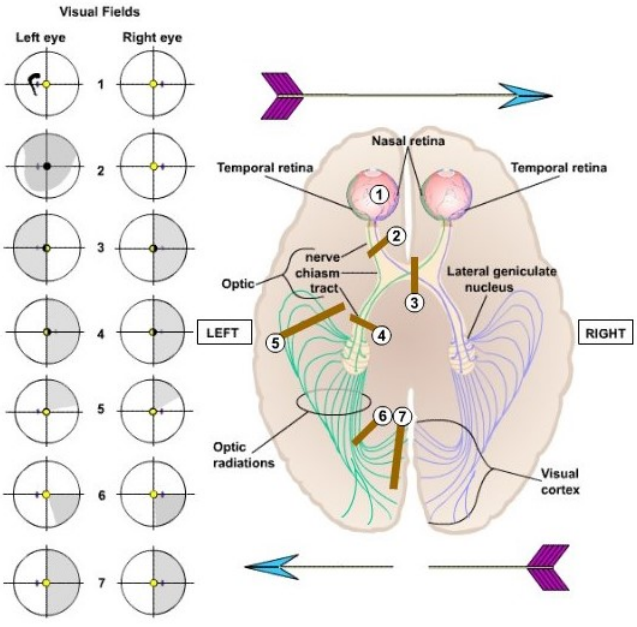


Figure 3.7: Right and Left Hemispheres Effect

Within those two hemispheres, each can be divided into four sections or lobes :

1. The frontal lobe thinks and gives orders to the muscles.
2. The parietal lobe is located at the top of the head and interprets sensations received by the skin.
3. The occipital lobe is at the base of the skull and analyzes images. It contains no less than thirty visual areas.
4. The temporal lobe decodes sounds and plays a role in memory.

These four sections are mirrored in each hemisphere, although some areas are specific to one hemisphere. For example, melody recognition and understanding emotions in the voice are localized in the right hemisphere, whereas centers for calculation, sentence comprehension, symbols, and language production are in the left hemisphere.

The link between these two hemispheres is the corpus callosum. Although each part of the brain focuses on a specific task, they collaborate to produce the necessary behaviors.

The visual cortex is made up of about thirty different regions or cortical areas. All of these areas work together to create a single, clear image. However, we can only define and understand the role of some of them.

The first, V1, is the primary visual area. It is the most important part because without it, we would be blind. Information coming from the retina through the optic nerve arrives here. In fact, each part of the retina has a "geographically" accurate reflection on the primary visual cortex. This area analyzes information on form, color, and movement and then distributes it to other areas of the visual cortex (such as V2, V3, V4, V5).

Area V2 receives a lot of information and further refines the data passed on by V1. Processes contours such as geometric shapes, orientations (horizontal and vertical), textures (smooth, rough, etc.), and colors.

At the same time, other areas receive information from V1, process it in parallel, but each focuses on its specialty :

- Area V3 analyzes moving shapes and evaluates distances.
- Area V4 handles color processing and stationary shapes.
- Finally, area V5 is responsible for the perception of movements such as direction and speed.

Most of the 30 visual areas identified are located in the occipital region (the back of the brain). However, some are found in the parietal regions (sides and top of the skull), which are thought to be involved in object localization. The temporal visual areas, on the other hand, are believed to be dedicated to object identification. Despite this, it remains difficult to understand how all these regions coordinate their functions to construct clear images.

Figure 3.8 shows how visual information is interpreted by the visual cortex regions from V1 to V5 and how they are finally devided into the "what" and "where" pathways of the brain.

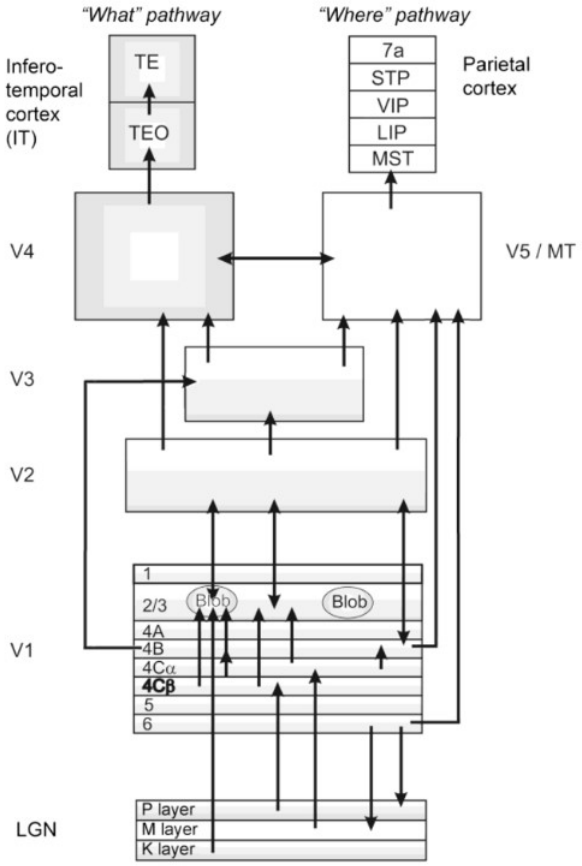


Figure 3.8: Schematic representation of the primate visual pathways

3.3 Image Processing

Before we delve into the modelisation of the human visual system for the feature extraction methods we will be using, we will introduce some basic image processing techniques for the preprocessing, segmentation, and normalization of the Iris

3.3.1 Preprocessing and Segmentation

3.3.1.1 Thresholding

Thresholding is a fundamental image processing technique used to segment an image into different regions by separating objects from the background based on intensity values. The primary goal is to create a binary image where specific pixels are designated as foreground (object) and others as background, thereby simplifying the analysis of the image.

The core idea of thresholding involves selecting a threshold value T. Pixels in the image are classified as follows:

- If the intensity of a pixel $I(x, y)$ is greater than the threshold T, it is set to one (white), indicating the presence of an object.

- If the intensity $I(x, y)$ is less than or equal to T, it is set to zero (black), indicating the background.

The mathematical representation for thresholding can be expressed as:

$$I_{thresh}(x, y) = \begin{cases} 1 & \text{if } I(x, y) > T \\ 0 & \text{if } I(x, y) \leq T \end{cases} \tag{3.1}$$

where:

- $I_{thresh}(x, y)$ is the output binary image,
- $I(x, y)$ is the input intensity value of the pixel at coordinates (x, y),
- T is the selected threshold value.

Thresholding can be categorized into two main types based on how the threshold is determined:

- **Static Thresholding:** In static thresholding, a global threshold T is chosen, which applies uniformly across the entire image. This threshold is determined before processing and remains fixed during segmentation. It works well when the intensity distributions of the foreground and background are distinctly separated.
- **Adaptive Thresholding:** In contrast, adaptive thresholding calculates the threshold dynamically based on local characteristics of the image. The threshold value varies for different regions, allowing for better segmentation in images with varying illumination or noise.

  In this approach, the threshold T for a pixel (x, y) can be determined using local statistics of pixel intensities within a neighborhood $N(x, y)$. A common method to determine the adaptive threshold is:

$$T(x, y) = \text{mean}(I(N(x, y))) - C \tag{3.2}$$

  where:

  - C is a constant value used to fine-tune the thresholding process,
  - $\text{mean}(I(N(x, y)))$ calculates the mean intensity of the neighborhood surrounding the pixel (x, y).

3.3.1.2 Histogram Equalization

The histogram of an image is the distribution of the intensity levels present in the image. When the distribution is not uniform, resulting in a bad image brightness, we need to transform the histogram in order to improve the image quality.

So Histogram Equalization is a technique in image processing used to improve the contrast of an image by effectively spreading out the most frequent intensity values. This process helps to enhance the visibility of features in the image.

The main goal of histogram equalization is to transform the intensity distribution of an image so that it approximates a uniform distribution. This leads to a more balanced representation of intensity values.

1. The histogram of an image represents the frequency of occurrence of each intensity value. For an image **I** with intensity values in the range **[0,L - 1]** (where **L** is the number of intensity levels, for exemple 256 for an 8 bit grayscale image, and L-1 represents then the maximum intensity level), the histogram can be computed as :

$$h(r_k) = \sum_{i=1}^{M} \sum_{j=1}^{N} \delta(I(i,j) - r_k) \tag{3.3}$$

   where:

   - $h(r_k)$ is the histogram value for intensity r_k,
   - M and N are the dimensions of the image,
   - δ is the Dirac delta function, which counts occurrences of intensity r_k.

2. the Probability Density Function (PDF) $p(r_k)$ is computed by normalizing the histogram :

$$p(r_k) = \frac{h(r_k)}{MN} \tag{3.4}$$

3. the Cumulative Distribution Function (CDF) $c(r_k)$ is obtained by summing the PDF values :

$$c(r_k) = \sum_{j=0}^{k} p(r_j) \tag{3.5}$$

   This gives the cumulative probability of intensity values up to r_k.

4. The intensity values are then mapped to new values using the CDF, typically scaled to the range of intensity levels. The new mapped intensity value s_k for each intensity r_k is calculated as:

$$s_k = \text{round}\left((L-1) \cdot c(r_k)\right) \tag{3.6}$$

   This step transforms the intensity values based on the cumulative distribution.

5. The final step is to construct the equalized image I_{eq} by replacing each pixel intensity $I(i,j)$ in the original image with its corresponding new intensity value s_k:

$$I_{eq}(i,j) = s_k I(i,j) \tag{3.7}$$

   where s_k is the mapped value from the previous step.

3.3.1.3 Canny Filter

The Canny filter is a multi-stage edge detection algorithm that is considered one of the most effective for detecting edges in images. the Canny filter is an analytic edge detection operator that aims to have a well-localized edge point and a single edge point response. It aims to identify a wide range of edges while reducing noise. The Canny edge detector follows several steps :

1. **Noise Reduction with Gaussien Filter :** The image is smoothed using a Gaussian filter to reduce noise :

   Gaussian Filter is a 2D convolution operator used to remove Gaussian noise and is more effective at smoothing images. The degree of smoothing is determined by the standard deviation of the Gaussian distribution.

   The 2D Gaussian Distribution formula :

$$F(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3.8}$$

   Where :

   σ : the standard deviation of the distribution.

   (x,y) : the coordinates of the pixel in question.

2. **Gradient Calculation :** The Sobel filter is applied to the smoothed image to compute the gradients G_x and G_y and their magnitude G :

   The Sobel filter is an edge detection operator used in image processing to find the gradient of the image intensity at each pixel. It helps identify edges by emphasizing areas of high spatial frequency, making it particularly effective for detecting horizontal and vertical edges edges in images.

   The Sobel filter consists of two 3x3 convolution kernels, one for detecting horizontal edges and the other for vertical edges :

   Horizontal Sobel kernel G_x:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.9}$$

   Vertical Sobel kernel G_y:

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{3.10}$$

   The gradients in the x and y directions can be calculated as follows:

$$G_x(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(x+i, y+j) \cdot G_x(i,j) \tag{3.11}$$

$$G_y(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(x+i, y+j) \cdot G_y(i,j) \tag{3.12}$$

   Where:
   - $G_x(x,y)$ and $G_y(x,y)$: Represent the gradient magnitudes in the horizontal and vertical directions, respectively.
   - $I(x,y)$: Denotes the original intensity value of the pixel at coordinates (x,y).

   To calculate the overall gradient magnitude, the Sobel filter combines these two gradients:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{3.13}$$

The direction of the gradient can also be determined:

$$\theta(x, y) = \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right) \tag{3.14}$$

3. **Non-Maximum Suppression :** This step thin the edges by preserving only the local maxima in the gradient direction, thus reducing the thickness of the detected edges.

   The previously calculated angle θ in 3.14 is typically quantized into four main directions (0°, 45°, 90°, and 135°) to simplify comparisons. For each pixel (x,y), Determine the neighboring pixels in the direction of θ, If G(x,y) is not greater than the gradients of its neighboring pixels, it is set to zero (suppressed).

4. **Double Thresholding :** Two threshold values, τ_{low} and τ_{high}, are used to classify pixels as strong, weak, or non-edges. Pixels with gradient magnitudes above τ_{high} are considered strong edges, while those between τ_{low} and τ_{high} are weak edges.
   The classification can be represented mathematically as:

$$\text{Edge Type} = \begin{cases} \text{Strong} & \text{if } G(x, y) > \tau_{high} \\ \text{Weak} & \text{if } \tau_{low} < G(x, y) \leq \tau_{high} \\ \text{Non-edge} & \text{if } G(x, y) \leq \tau_{low} \end{cases} \tag{3.15}$$

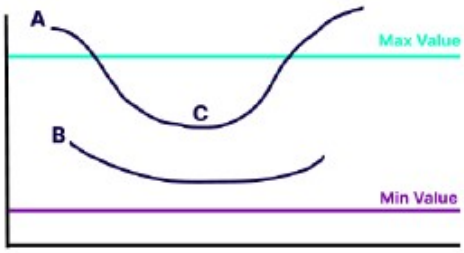


Figure 3.9: Double Thresholding

5. **Edge Tracking by Hysteresis :** In this final step, weak edges are analyzed to determine if they are connected to strong edges. This process ensures that true edges are retained while reducing noise.

   Connecting Weak Edges : For each weak edge pixel, if it is connected to any strong edge pixel (i.e., it is adjacent to a strong edge pixel), it is retained as an edge; otherwise, it is discarded.

   Connection Check : This can be visualized using an 8-connected neighborhood. If a weak edge pixel (x, y) has a strong edge pixel in its 8-connected neighborhood, it is retained:

$$\text{Retain}(x, y) = \begin{cases} \text{True} & \text{if } \exists (s, t) \in N_8(x, y) \text{ such that } G(s, t) > T_2 \\ \text{False} & \text{otherwise} \end{cases} \tag{3.16}$$

   Where $N_8(x, y)$ represents the 8-connected neighborhood of pixel (x, y).

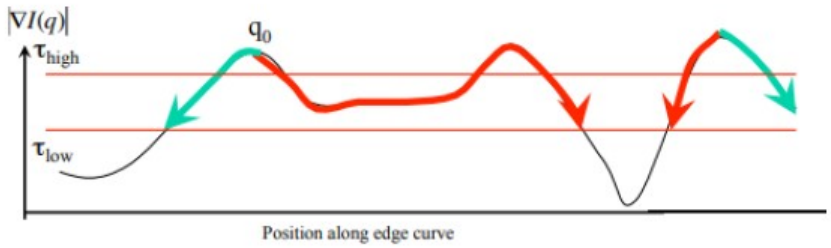


Figure 3.10: Edge Tracking by Hysteresis

Exemple of an application of the canny filter :

Figure 3.11: Edge Extraction Using Canny Filter

3.3.1.4 Laplacien of Gaussien (LoG)

The Laplacian of Gaussian (LoG) is a combined image processing technique used primarily for edge detection. It blends two operations: Gaussian smoothing to reduce noise, followed by the Laplacian operator to detect areas of rapid intensity change (edges). LoG is particularly useful for detecting edges that have been smoothed out to minimize noise.

The Laplacien of Gaussien consits of :

1. **Gaussian Smoothing :** The first step involves applying a Gaussian filter to the image to smooth it and reduce noise. The Gaussian Filter is seen in equation 3.8.
2. **cian Operator :** After smoothing, the Laplacian operator is applied to detect the edges. The Laplacian is a second-order derivative operator that highlights regions of rapid intensity change, commonly used for edge detection. It is defined as :

$$\nabla^2 I(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{3.17}$$

Where:

- $I(x, y)$ is the image intensity at pixel (x, y).
- ∇^2 represents the Laplacian operator.

The Laplacian responds to regions where the intensity gradient is changing rapidly, which often corresponds to edges in the image.

3. **Laplacian of Gaussian (LoG) :** The Laplacian of Gaussian applies both operations in one step. Instead of applying the Gaussian smoothing and then the Laplacian separately, LoG combines them into a single filter:

$$\text{LoG}(x, y) = \nabla^2 \left[\frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2} \right) \right] \tag{3.18}$$

This equation represents the convolution of the image with the Laplacian of the Gaussian function. The result is an edge-enhanced version of the image, where edges appear as zero-crossings.

4. **Zero-Crossings and Edge Detection :** In the LoG method, edges are detected by finding the zero-crossings of the filtered image. Zero-crossings occur where the second derivative changes sign (from positive to negative or vice versa). These crossings indicate the presence of an edge, making it possible to detect object boundaries and other prominent features in the image.

3.3.1.5 Hough Transform

The Hough Transform which is a segmentation method is a technique that can be used to isolate features of a particular shape within an image. It requires that the desired features be specified in some parametric form.

The classical Hough Transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. Despite its domain restrictions, the classical Hough Transform retains many applications, as most manufactured parts contain feature boundaries that can be described by regular curves.

The main advantage of the Hough Transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise [89].

Steps of the Hough Transform

1. **Edge Detection:** Before applying the Hough Transform, the image is typically processed using edge detection techniques (e.g., Canny edge detector) to identify points that likely belong to the desired shapes.

2. **Parameter Space Representation:** Each edge point (x, y) is transformed into the Hough space by calculating all possible values of (ρ, θ) that could correspond to lines passing through that point. This is done by iterating through a range of θ values:

$$\rho = x\cos(\theta) + y\sin(\theta)$$

For each (x, y), this generates a curve in the (ρ, θ) space.

3. **Accumulator Array:** An accumulator array (or Hough space) is used to store the count of intersections. Each cell in this array corresponds to a specific (ρ, θ) pair. As edge points are processed, votes are cast in the accumulator array for each possible line defined by those points.

4. **Peak Detection:** After processing all edge points, local maxima in the accumulator array are identified. Each peak corresponds to a parameter pair (ρ, θ) that indicates a detected line in the image. The higher the count in a cell, the more likely it represents a true line.

5. **Line Reconstruction:** Finally, the detected lines can be reconstructed in the original image using the (ρ, θ) values. The Cartesian coordinates of the lines can be derived back from the polar coordinates:

$$y = \frac{\rho - x\cos(\theta)}{\sin(\theta)} \tag{3.19}$$

6. **Extensions to Other Shapes**

   While the Hough Transform is often associated with line detection, it can also be extended to detect other shapes, such as:

   - **Circles:** The equation for a circle in polar coordinates can be expressed as:

$$(x - a)^2 + (y - b)^2 = r^2 \tag{3.20}$$

     This can be transformed into a three-dimensional accumulator space where each point (x, y) votes for all possible circles defined by center (a, b) and radius r.
   - **Ellipses and Other Parametric Shapes:** The Hough Transform can be adapted to detect more complex shapes by extending the parameter space accordingly.

3.3.1.6 Daugman's Rubber Sheet Model

The primary goal of Daugman's rubber sheet model is to normalize the iris image so that variations in size and position due to changes in pupil size, rotation, and perspective can be accounted for. By transforming the iris into a rectangle, it simplifies the process of matching and analyzing the iris patterns.

1. **Iris Localization:** First, identify the circular boundaries of the iris and pupil. Let:

   - R_p = Radius of the pupil
   - R_i = Radius of the iris

   The center of the pupil can be denoted as

$$C = (x_c, y_c).$$

2. **Polar Coordinate Transformation:** For each point on the iris, we can convert Cartesian coordinates to polar coordinates:

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad \text{(distance from center)}$$
$$\theta = \tan^{-1}\left(\frac{y - y_c}{x - x_c}\right) \quad \text{(angle with respect to the center)}$$

Where (x, y) are the coordinates of a point on the iris.

3. **Mapping to a Rectangular Domain:** The goal is to normalize the iris pattern into a rectangular shape, where:

   - The horizontal axis represents the angular coordinate θ.
   - The vertical axis represents the radial coordinate r.

   The transformation can be expressed as:

   - **Normalized Width (for θ):**

   $$\Theta \in [0, 2\pi] \quad \text{(mapped to a fixed width)}$$

   - **Normalized Height (for r):**

   $$R \in [R_p, R_i] \quad \text{(mapped to a fixed height)}$$

4. **Rectangular Mapping:** The points on the iris are transformed to rectangular coordinates (u, v):

$$u = \frac{\theta}{2\pi} \cdot W \quad \text{(where } W \text{ is the fixed width of the rectangle)}$$
$$v = \frac{r - R_p}{R_i - R_p} \cdot H \quad \text{(where } H \text{ is the fixed height of the rectangle)}$$

   Here:

   - u represents the normalized width corresponding to the angle θ.
   - v represents the normalized height corresponding to the radius r.

5. **Final Normalization:** After transforming the iris points into rectangular coordinates, the resulting normalized iris pattern $I_{norm}(u, v)$ can be constructed:

$$I_{norm}(u, v) = I(x, y) \quad \text{for each point mapped from circular to rectangular coordinates}$$

   Where $I(x, y)$ is the original intensity value of the pixel at (x, y) in the iris image.

3.3.1.7 Enhancement Using Filtering

Filtering involves either allowing or eliminating specific frequency components within a signal, while spatial filtering is a technique in which a specific operation is applied to a surrounding area of pixels to produce a new pixel, positioned in the center of that area, with its value derived from the filtering process. As the filter, also referred to as a mask or kernel, moves across each pixel in the image, it results in the creation of a new image, known as the filtered image.

Median Filter :
The Median filter is a type of nonlinear filter often used in image processing to remove noise while preserving edges. It operates by replacing the value of a pixel with the median value of the intensity levels from its neighboring pixels. This process is particularly effective in reducing both salt-and-pepper noise and impulse noise.

The formula for the median filter can be expressed as follows:

$$f(x, y) = \text{median}(I(s, t)) \quad \text{where } (s, t) \in S_{xy} \tag{3.21}$$

Where :

- $f(x, y)$: Represents the intensity of the new pixel at the coordinates (x, y).
- S_{xy}: The set of neighboring pixels surrounding the center pixel (x, y).
- (s, t): The coordinates of the pixels within the neighborhood S_{xy}.
- $I(s, t)$: The intensity value of the pixel located at (s, t).

In contrast to other linear filters, such as the Gaussian filter, the Median filter significantly reduces impulse noise while maintaining the overall sharpness of the image, making it a popular choice in various applications.

Top and Bottom Hat Filters :

- **The Top-Hat filter**, also known as the white top-hat transform, is a morphological operation used in image processing to enhance bright features within an image. This filter effectively highlights small, bright regions that stand out against a darker background, making it particularly useful for extracting small objects.

  The formula for the Top-Hat filter can be expressed as follows:

$$f_{\text{top}}(x, y) = I(x, y) - \gamma(I)(x, y) \tag{3.22}$$

  Explanation:

  - $f_{\text{top}}(x, y)$: Represents the output intensity at the pixel coordinates (x, y) after applying the Top-Hat filter.
  - $I(x, y)$: Denotes the original intensity value of the pixel at coordinates (x, y).
  - $\gamma(I)$: Refers to the morphological opening of the image I using a specified structuring element, defined as:

$$\gamma(I) = (I \circ B) \oplus B \tag{3.23}$$

where:

- $\circ$ represents morphological erosion, which removes pixels on object boundaries, defined mathematically as:

$$(I \circ B)(x, y) = \max\{(x', y') \mid (x', y') \in B \text{ and } I(x + x', y + y') > 0\} \tag{3.24}$$

This operation effectively shrinks the bright regions in the image.

- $\oplus$ represents morphological dilation, which adds pixels to the boundaries of objects in an image, defined mathematically as:

$$(I \oplus B)(x, y) = \max\{(x', y') \mid (x', y') \in B \text{ and } I(x - x', y - y') > 0\} \tag{3.25}$$

This operation expands the bright regions in the image.

The Top-Hat filter effectively isolates bright spots in an image, making it particularly useful in applications such as medical imaging and industrial inspection.

- **The Bottom-Hat filter**, also known as the black bottom-hat transform, is another morphological operation designed to enhance dark features in an image. This filter identifies small, dark regions that are darker than their surroundings, making it useful for extracting small objects against a lighter background.

The formula for the Bottom-Hat filter is given by:

$$f_{\text{bottom}}(x, y) = \gamma(I)(x, y) - I(x, y) \tag{3.26}$$

Explanation:

- $f_{\text{bottom}}(x, y)$: Represents the output intensity at the pixel coordinates (x, y) after applying the Bottom-Hat filter.
- $I(x, y)$: Denotes the original intensity value of the pixel at coordinates (x, y).
- $\gamma(I)$: Refers to the morphological closing of the image I using a specified structuring element, defined as:

$$\gamma(I) = (I \cdot B) \ominus B \tag{3.27}$$

where:

- $\cdot$ represents morphological dilation, defined previously, which adds pixels to object boundaries, effectively enlarging dark regions.
- $\ominus$ represents morphological erosion, defined previously, which removes pixels from the boundaries of objects in the image.

The Bottom-Hat filter effectively highlights dark regions against a lighter background, making it useful for tasks such as shadow detection and text extraction in images.

The Figure 3.12 is a 2D and 3D representations of Top Hat filter:

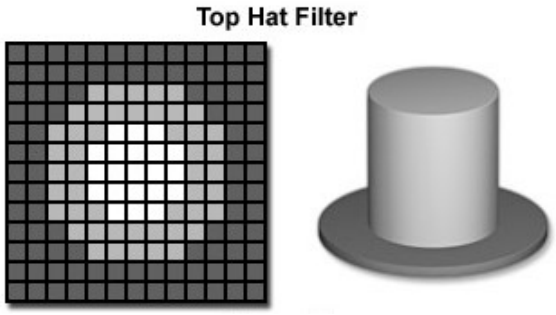


Figure 3.12: Top Hat Filter

3.3.2 Features Extraction

3.3.2.1 Gabor Filter

A Gabor filter is a linear filter used in image processing for edge detection, texture classification and features extraction.

It is a band-pass filter, i.e. it passes frequencies in a certain band and attenuates the other frequencies outside such band. A Gabor filter is essentially sinusoidal functions modulated by a Gaussian envelope. The mathematical representation of a 2D Gabor filter can be expressed by formula 3.28 in spatial domain :

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left[i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right] \tag{3.28}$$

With :

$$x' = x\cos\theta + y\sin\theta \tag{3.29}$$

$$y' = y\cos\theta - x\sin\theta \tag{3.30}$$

Where:

(x,y) : the coordinates of the pixel,

λ : the wavelength of the plane wave,

θ : he orientation of the normal to the parallel stripes of the Gabor function,

σ : the standard deviation of the Gaussian component,

γ : the aspect ratio that defines the ellipticity of the function support,

ψ : the phase of the plane wave.

Multiple filters can be generated by adjusting the parameters $\lambda, \theta, \sigma, \psi$ and γ. A collection of Gabor filters, known as a Gabor filter bank, consists of various filters that utilize different parameter settings. By altering these Gabor parameters, one can extract a range of low-level

features from the original image through the convolution process, thereby establishing the Gabor filter as a highly effective instrument for feature extraction.

Implementation of Gabor Filters

Convolution: To apply a Gabor filter to an image, the filter is convolved with the image. This process highlights features that match the characteristics of the filter. The convolution operation can be represented mathematically as:

$$I_{\text{filtered}}(x, y) = (I * g)(x, y) = \int \int I(u, v)g(x - u, y - v)\, du\, dv \tag{3.31}$$

Where I is the input image, and g is the Gabor filter.

Magnitude and Phase: Since Gabor filters produce complex-valued outputs, the magnitude and phase can be computed for further analysis:

Magnitude:

$$|g(x, y)| = \sqrt{\text{Re}(g(x, y))^2 + \text{Im}(g(x, y))^2} \tag{3.32}$$

Phase:

$$\phi(x, y) = \tan^{-1}\left(\frac{\text{Im}(g(x, y))}{\text{Re}(g(x, y))}\right) \tag{3.33}$$

3.3.2.2 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is a mathematical tool used to analyze and represent signals or images at multiple levels of resolution. Applied in signal processing, image compression, denoising, and feature extraction. Unlike the Fourier transform, which breaks down a signal into sinusoids of different frequencies, the wavelet transform uses small, localized waveforms known as wavelets that can capture both frequency and location information simultaneously, making it ideal for non-stationary signals (e.g., images or time-varying signals).

Wavelets: A wavelet is a small wave-like function that is localized in both time (or space) and frequency. It has both a scaling property and a shifting property, which allows it to capture details at different resolutions.

The mother wavelet is the base wavelet from which all other wavelets are derived through scaling and translation. Mathematically, a wavelet can be represented as:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t - b}{a}\right) \tag{3.34}$$

Where:
- ψ is the mother wavelet,
- a is the scaling factor (controls the width of the wavelet),
- b is the translation factor (controls the position of the wavelet),
- t is time or space.

Multi-Resolution Analysis: DWT performs multi-resolution analysis by decomposing a signal (or image) into different levels of detail. This is done using two functions:

- *Scaling Function (Low-Pass Filter):* Extracts the coarse, low-frequency components (approximation).
- *Wavelet Function (High-Pass Filter):* Extracts the fine, high-frequency details (details).

This decomposition process involves convolving the signal with the scaling and wavelet functions, followed by down-sampling.

For a 1D signal $f(t)$, the DWT at each level involves two outputs:
- Approximation Coefficients (A): Captures the low-frequency components (smooth part of the signal).
- Detail Coefficients (D): Captures the high-frequency components (edges, sharp transitions).

For a 2D image, the DWT results in four sub-bands:
- *LL:* Approximation (low-pass in both directions).
- *LH:* Horizontal details (low-pass in the vertical direction, high-pass in the horizontal direction).
- *HL:* Vertical details (high-pass in the vertical direction, low-pass in the horizontal direction).
- *HH:* Diagonal details (high-pass in both directions).

DWT Decomposition: The DWT decomposition of an image $I(x, y)$ can be written as a series of convolutions:

$$A_{i+1} = \sum_{x,y} I(x, y) \cdot \phi_{i+1}(x, y) \tag{3.35}$$

$$D_{i+1} = \sum_{x,y} I(x, y) \cdot \psi_{i+1}(x, y) \tag{3.36}$$

Where:
- A_{i+1} represents the approximation coefficients at level $i + 1$,
- D_{i+1} represents the detail coefficients at level $i + 1$,
- ϕ_{i+1} is the scaling function at level $i + 1$,
- ψ_{i+1} is the wavelet function at level $i + 1$,
- x, y are pixel coordinates.

At each step, the decomposition provides a finer level of detail.

Reconstruction: The original signal or image can be reconstructed by reversing the decomposition process using inverse DWT (IDWT), where approximation and detail coefficients are combined to restore the original information.

The Figure 3.13 shows some common wavelets :

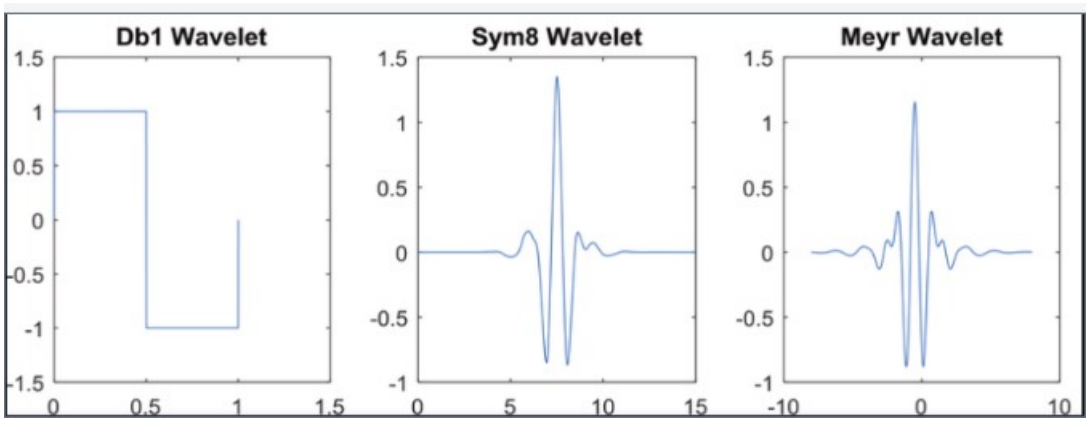


Figure 3.13: Examples of different types of wavelets

In order to know how efficient our HVS based models (DWT and Gabor Filters) are, we decided to use 2 non HVS based methods : one consisting of only Frequency domain (based on cosine functions) which is the Discrete Cosine transform, while the other one operates purely in the spatial domain by analyzing gradients (Histogram of Gradients).

3.3.2.3 Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is a mathematical technique widely used in signal and image processing for transforming data from the spatial domain to the frequency domain. DCT is particularly well-known for its application in image and video compression, such as JPEG and MPEG standards. Like other transforms, the DCT helps to separate the image into parts of varying importance by concentrating energy into the lower frequencies, making it easier to discard higher-frequency components without significant loss of image quality. Discrete Cosine Transform include the following steps ;

1. **Transforming Data :** The DCT converts a signal (such as an image) from its spatial representation (pixel values) into a frequency representation (coefficients). It operates similarly to the Fourier Transform, but only uses cosine functions. The DCT expresses the input data as a sum of cosine functions oscillating at different frequencies.

   The most common form is the 2D DCT for images, which operates on image blocks (such as 8×8 blocks), converting pixel values into frequency coefficients.

   The 2D DCT for an $N \times N$ image or block is defined as:

$$C(u,v) = \frac{1}{4}\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{3.37}$$

   Where:

   - $I(x,y)$ represents the intensity of the pixel at coordinates (x,y).
   - $C(u,v)$ is the DCT coefficient at frequency position (u,v).
   - N is the block size.
   - $\alpha(u)$ and $\alpha(v)$ are scaling factors defined as:

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } u = 0 \\ \frac{\sqrt{2}}{\sqrt{N}} & \text{if } u > 0 \end{cases}$$

In an image, low-frequency components correspond to slow variations in intensity, while high-frequency components represent rapid changes (such as edges and fine details). The DCT has the property of concentrating most of the signal's energy into the low-frequency components (top-left corner of the DCT matrix). This makes it ideal for compression, as the less important high-frequency components can be discarded with minimal visual impact.

3.3.2.4 Histogram of Gradients (HoG)

The Histogram of Oriented Gradients (HOG) is a feature extraction technique that focuses on capturing the shape and structure of objects by analyzing the gradient orientation and magnitude within localized regions of an image.

The first step in the HOG algorithm is to compute the gradients of the image. The gradient describes the rate of change in pixel intensity in a particular direction. These gradients highlight edges, contours, and corners in the image, which are essential for describing the object's shape. For an image I, the gradient along the x-axis and y-axis can be computed using simple filters:

$$G_x(x, y) = I(x + 1, y) - I(x - 1, y)$$
$$G_y(x, y) = I(x, y + 1) - I(x, y - 1)$$

These gradients give the magnitude and orientation at each pixel.

After computing the gradients, the next step is to create a histogram of gradient orientations within small spatial regions, known as cells. Each cell typically contains a group of pixels (e.g., 8x8 pixels). For each pixel in the cell, the gradient's orientation is calculated (usually in degrees between 0 and 180), and its magnitude is used as the weight for that orientation in the histogram. The histogram bins represent different angles (e.g., 0°, 20°, 40°, etc.), and each bin accumulates the gradient magnitudes of pixels with orientations that fall within that bin's range.

To make the feature extraction robust to changes in illumination and contrast, the histogram values are normalized across overlapping regions of the image, called blocks. A block consists of several cells (e.g., 2x2 cells), and the histograms from these cells are normalized together. This step ensures that the HOG features are invariant to changes in lighting. The normalization can be done using techniques like L2-norm or L1-norm:

$$v' = \frac{v}{\|v\|_2^2 + \epsilon^2}$$

where v is the unnormalized vector of histogram values, and ϵ is a small constant to avoid division by zero.

The normalized histograms from all the blocks are then concatenated to form a feature vector. This vector serves as the HOG descriptor, which captures the shape and structure of the object in the image.

3.3.3 Features Selection

3.3.3.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely used statistical technique for dimensionality reduction, feature extraction, and data visualization. It transforms high-dimensional data into a lower-dimensional form by identifying the directions (called principal components) that capture the most variance in the data. PCA is especially useful when dealing with datasets with many variables, as it helps reduce the complexity while preserving the essential structure of the data.

How PCA Works

1. Data Centering : The input data is first centered by subtracting the mean of each feature from the corresponding values. This ensures that the data has a mean of zero along each dimension, which is necessary for PCA to identify the direction of maximum variance.

$$X_{\text{centered}} = X - \bar{X} \tag{3.38}$$

Where:

- X is the original data matrix.
- $\bar{X}$ is the mean of the data.

2. Covariance Matrix Calculation: PCA computes the covariance matrix of the centered data, which quantifies how much the features vary together. The covariance matrix is defined as:

$$C = \frac{1}{n-1} X_{\text{centered}}^T X_{\text{centered}} \tag{3.39}$$

Where:

- C is the covariance matrix.
- n is the number of data samples.

3. Eigenvalue and Eigenvector Decomposition : PCA performs eigenvalue decomposition on the covariance matrix. The eigenvectors represent the principal components, and the corresponding eigenvalues indicate the amount of variance explained by each component. The principal components are ordered by the magnitude of the eigenvalues, with the first component capturing the largest variance.

$$Cv_i = \lambda_i v_i \tag{3.40}$$

Where:

- v_i is the i-th eigenvector (principal component).
- λ_i is the i-th eigenvalue (variance explained by that component).

4. Projection onto Principal Components : Once the principal components are identified, the data is projected onto these new axes, transforming the original data into a lower-dimensional space. The transformed data is given by:

$$Z = X_{\text{centered}} V_k \tag{3.41}$$

Where:

- Z is the transformed data in the reduced dimensional space.
- V_k is a matrix of the top k eigenvectors (principal components).

Key Features of PCA

- **Dimensionality Reduction:** PCA reduces the number of features while retaining the most important information, making it particularly useful in high-dimensional datasets.
- **Variance Maximization:** PCA identifies directions where data varies the most, making it effective in capturing the underlying structure of the data.
- **Uncorrelated Features:** The principal components are orthogonal, meaning they are uncorrelated with each other, ensuring that each captures a unique aspect of the variance in the data.

3.4 Modelisation of SVH

3.4.1 Luminance Perception

Understanding the process of visual perception is a key consideration when modeling the Human Visual System (HVS). The perception of an image region generates three types of sensations: hue, saturation, and brightness. Hue and saturation are linked to the chromaticity of the observed area, while brightness reflects perceived luminance. Several processes contribute to the perception of a wide luminance dynamic range, from $10^{-2}\,\text{cd/m}^2$ to $10^{5}\,\text{cd/m}^2$. While cones require a certain level of brightness, rods react even to very low luminance levels. This sensitivity to very low luminance is also ensured by spatial integration occurring at the level of ganglion cells.

Weber proposed one of the first quantitative models of luminance perception. Weber's law is modeled by the equation :

$$\frac{\Delta L}{L} = \text{Constant} \tag{3.42}$$

This law indicates that if a uniform background luminance L (referred to as the adaptation luminance) overlaps a stimulus (a disc of luminance $\Delta L + L$), the ratio $\frac{\Delta L}{L}$ remains nearly constant across a wide range of luminance.

Several studies confirmed these measurements, which determine detection thresholds based on background luminance when it is sufficiently high. However, some slight inconsistencies in low luminance were corrected by Moon and Spencer, resulting in the following expression :

$$\frac{\Delta L}{L} = \frac{C_\infty}{L}(0.465 + \sqrt{L})^2 \tag{3.43}$$

As luminance increases, the ratio $\frac{\Delta L}{L}$ tends toward the constant value C, which depends on the geometry and size of the stimulus. Weber's law, as refined by Moon and Spencer, demonstrates that the HVS undergoes a logarithmic-like transformation, as illustrated in Figure 3.14.

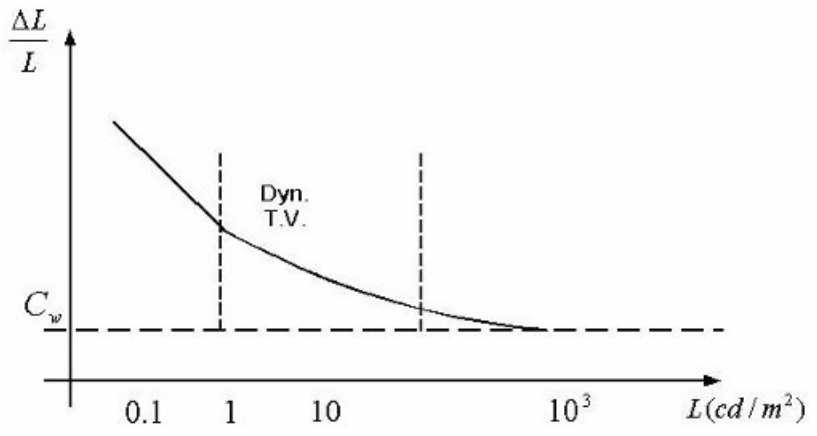


Figure 3.14: Moon and Spencer Law

3.4.2 Contrast

Contrast is the ratio between the local intensity of an image and its average intensity. There are many definitions of contrast, adapted to more or less complex stimuli. One of the most general Formulas of the contrast :

$$C = \frac{L_{\max} - L_{\min}}{L_{\max} + L_{\min}} \tag{3.44}$$

The local band-limited contrast involves decomposing the image into visual sub-bands and is defined as the ratio between the local luminance of a given sub-band and the local average luminance relative to this channel.

$$C_i(m,n) = \frac{L_i(m,n)}{\sum_{k=0}^{i-1} L_k(m,n)} \tag{3.45}$$

where i represents the i-th radial channel, and $\sum_{k=0}^{i-1} L_k(m,n)$ is the low-frequency signal relative to the i-th band.

3.4.3 Contrast Sensitivity Function (CSF)

The Contrast Sensitivity Function (CSF) describes the ability of the Human Visual System (HVS) to detect differences in luminance (contrast) between an object and its background across a range of spatial frequencies. It is a critical concept in visual perception that quantifies how sensitivity to contrast varies with spatial frequency (how fine or coarse a pattern is).

Spatial frequency refers to the level of detail in a visual stimulus, usually measured in cycles per degree (cpd), where a "cycle" is a pair of dark and light bars, and "per degree" refers to the visual angle they subtend on the retina. The CSF essentially captures how well humans perceive contrast at different levels of detail.

3.4.4 Masking

Signals with similar characteristics in the visual system are processed through the same pathways, interacting with each other and producing nonlinear effects, such as masking. Masking occurs when a stimulus's detection threshold is affected by the presence of a stronger signal with similar features[84]. A common model for this effect adjusts the threshold contrast of the stimulus based on the masking signal's characteristics. The model incorporates a weighting function, often modeled by a Gaussian distribution, which adjusts for the differences in frequency and angular characteristics between the stimulus and the masking signal. The model also accounts for changes in radial and angular bandwidth, which vary across different spatial frequencies.

3.4.5 Perceptual Channel Decomposition and Multi-Channel SVH Model

Spatial Vision Models (SVH models) describe how the Human Visual System (HVS) processes spatial information, particularly how we perceive spatial frequencies, patterns, and contrasts. Two primary models describe how the visual system processes information: Single-Channel and Multi-Channel Models.

Multi-Channel Models suggest that the HVS contains multiple independent channels or filters, each tuned to specific ranges of spatial frequencies. This model reflects a more biologically accurate representation of how the brain processes visual information, The internal retina, ganglion cells, the lateral geniculate body, and to some extent, the visual cortex are considered in this visual model. With different channels sensitive to different spatial details (low, medium and high frequencies) in an image. These channels work in parallel to interpret various spatial patterns.

Perceptual Channel Decomposition is a method used to analyze and represent visual information by breaking down an image into multiple components or channels based on how the human visual system (HVS) processes visual stimuli. Perceptual Channel Decomposition is derived from the works of Sénane and Le Callet[90]. This decomposition proposes a frequency tiling into seventeen channels, as illustrated in 3.15.

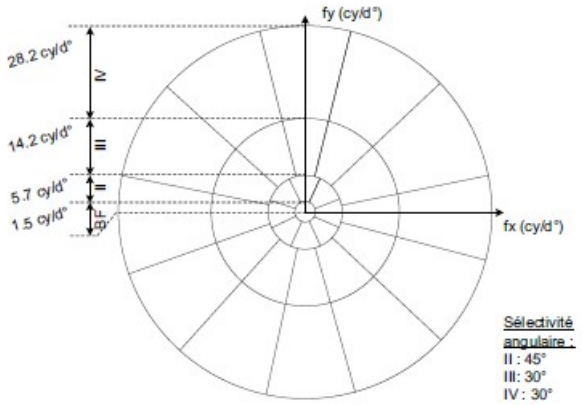


Figure 3.15: Frequency tiling of the Perceptual Channels Decomposition

Among the filters utilized in DCP, DOM filters and Fan filters are two prominent types. These filters are designed to extract specific directional features, enabling a more refined analysis of spatial patterns in images.

1. **DOM Filters :**
   DoM filters are radial band-pass filters derived from a base filter known as the Mesa filter. They are used to extract specific frequency components from images, particularly in the context of human visual perception.

   The Mesa filter is a circularly symmetric low-pass filter characterized by a Gaussian shape in the frequency domain. Its formulation is given by:

   $$M_0(u,v) = f_0 \cdot e^{-\frac{(u^2+v^2)}{2\sigma^2}} \cdot \Pi_{2f_0} \tag{3.46}$$

   Here, f_0 represents the cutoff frequency, σ is the standard deviation of the Gaussian, and Π_{2f_0} is the rectangular function that confines the filter to a certain range in the frequency domain.

   DoM filters are constructed by taking the difference between two Mesa filters with different cutoff frequencies. This results in a band-pass filter that allows specific frequency bands to pass through while attenuating others.

2. **FAN Filters :** Fan filters are angular band-pass filters that allow for the analysis of specific orientations in images. They are built by combining the outputs of multiple DoM filters or Mesa filters with different orientations.

3.4.6 Discrete Wavelete Transform and Perceptual Channel Decomposition

The wavelet transform shows similarities with the multi-channel organization of the human visual system. Like perceptual channel decomposition, the wavelet transform decomposes the image into several sub-bands. These sub-bands correspond to a limited frequency range and a limited set of orientations. Moreover, the content of each sub-band corresponds to a specific spatial location. However, there are differences between the DCP (decomposition in perceptual channels) and the DWT (discrete wavelet transform) :

- Separation and Dyadicity: The DWT is separable and dyadic, while the DCP is neither.
- Angular Selectivity: The DCP has angular selectivity of 45° or 30°, depending on frequency ranges, whereas the DWT has a fixed angular selectivity of 45° with diagonal sub-bands containing both 45° and -45° orientations, leading to potential interference issues.
- Filter Shape and Overflow: DWT filters can cause overlaps between horizontal, vertical, and diagonal sub-bands, affecting how contours are represented, especially in terms of masking effects.

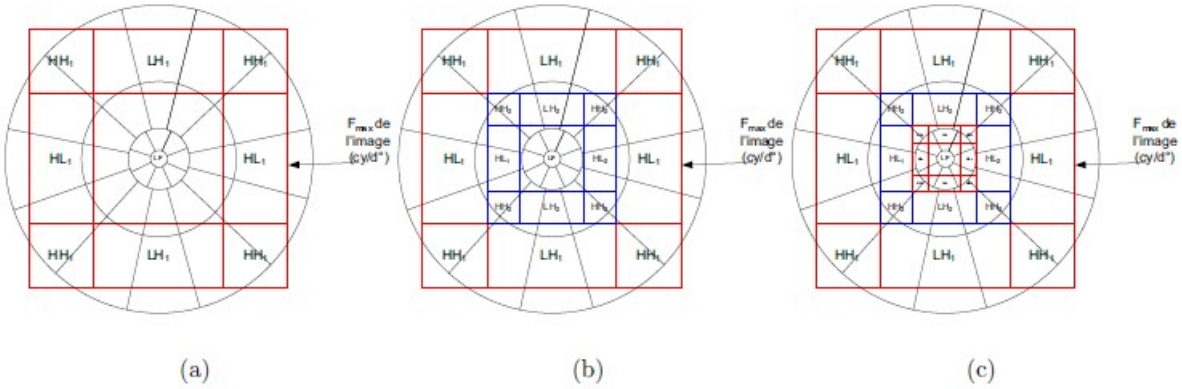


Figure 3.16: DCP and DWT Correspondance depending on DWT Decomposition Levels, a. one level. b. two levels. c. three levels

To maximize the correspondence between the DCP and our decomposition based on the DWT, the number of decomposition levels of the DWT is determined based on the maximum visible spatial frequency f_{max} in the image (i.e., the observation conditions). The frequency f_{max} is expressed in cycles per degree (cy/d°) by:

$$f_{\max} = 2f_s = 2(2\tan(0.5°) \cdot r_H) \tag{3.47}$$

where :
r represents the observation distance in terms of the image height, and H represents the height of the image in pixels and f_s The spatial sampling frequency of an image expressed in terms of pixels per degree of visual field.

The number of decomposition levels is then determined by increasing it until the low frequency of the DWT coincides with the low-frequency band (01.5 cy/d°) of the DCP, as illustrated in figure :

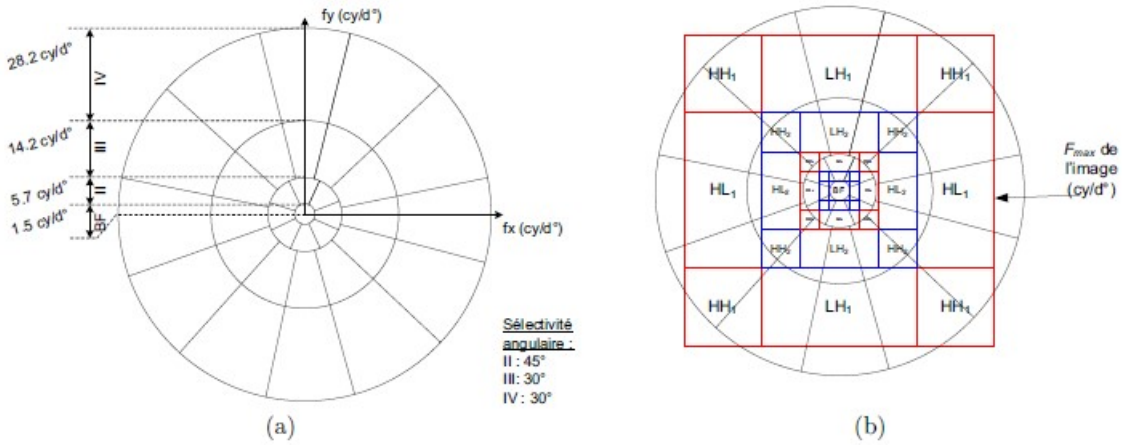


Figure 3.17: Approximation of DWT to DCP

3.5 Classification/Matching

The final step of iris recognition is classifying the features we obtained and then matching them with the features we have in our database, for this final step we use two methods : Deep Learning based methods which consists of CNNs and Machine Learning Based methods which consists of SVM, KNN and Random Forests.

3.5.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep learning architecture designed for processing structured grid data, such as images. Using Convolution layers scan over input data, identifying patterns and features such as edges, textures, and shapes at different levels. CNNs typically consist of layers like convolutional layers, pooling layers, and fully connected layers, often paired with non-linear activation functions to model complex relationships in the data.

1. The 2D convolution layer, or conv2D, is a core component used extensively in convolutional neural networks. In this layer, filters consist of multiple kernels, each one mapped to a specific input channel of the layer. Each kernel operates independently to fulfill its role in the convolution process.

   The conv2D layer performs a 2D convolution over input data, often images or feature maps, by processing as follows[91]:

   - Filters : The layer applies several filters, each containing a set of kernels. The number of filters directly corresponds to the number of output channels generated by the layer.
   - Convolution operation : Convolution is performed by sliding each kernel over the input tensor, calculating an element-wise product between the kernel and a corresponding region of the input. These products are summed at each location to form a single output value.

- Activation: A non-linear activation function is applied to the summed output, introducing non-linearity to the network. A commonly used activation function is the Rectified Linear Unit (ReLU), which is defined as :

$$ReLU(x) = max(0, x); \forall x \in \mathbb{R}. \tag{3.48}$$

- Output : After the convolution operation and activation function are applied, the result is a new 3D tensor, or feature map, with dimensions influenced by the number of filters and the spatial dimensions of the input. This feature map represents the output of the conv2D layer, encoding learned features from the input image.

2. The pooling layer performs a down-sampling operation by reducing the spatial dimensions of the input, focusing on width, height, and stride. This operation helps decrease the overall number of parameters and computational load, while still retaining the key features of the data. In practice, the input is divided into small, non-overlapping square regions, separated by a defined stride to avoid excessive information loss. Although the number of output feature maps stays the same as the input, the spatial resolution of each feature map is reduced.

   Pooling is typically carried out in one of two ways: max-pooling or average pooling. Max-pooling selects the highest value within the defined region, whereas average pooling computes the average of the values in that area. Max-pooling is more frequently used in practice, as it generally provides better results than average pooling. [92].

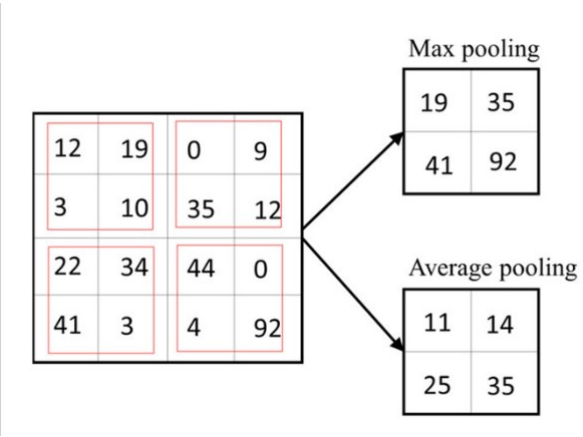


Figure 3.18: Max and AVG Pooling

3. The flatten layer is typically used as a transition layer between convolutional layers and the prediction block in deep learning models. It reshapes the input tensor into a one-dimensional vector. It takes the multidimensional input, such as an image or a feature map, and flattens it into a single continuous vector without changing the content.

   The purpose of the flatten layer is to convert the spatial information present in the input tensor into a format that can be passed to the neural network for further processing.

4. The Convolution operation as well as the Pooling or any other operation based on a kernel, are influenced by parameters like *Stride* and *Padding*.

   The **stride** determines the step size of the kernel while sliding over the input, affecting the size of the output, in other words, stride is the number of pixels shifts over the input matrix [92].

The **Padding** which refers to the addition of extra elements (typically zeros) around the edges of an input tensor or image, can be added to the input to preserve spatial dimensions or avoid border effects.

5. Dropout is a regularization method employed during the training phase to address overfitting by randomly deactivating a specified number of neurons. This strategy guarantees that no individual neuron excessively influences the learning process [92].

6. Adam Optimizer is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. Its name is derived from adaptive moment estimation.

7. In the classification problems that have a multi-class single label, the Sparse Categorical Cross Entropy (SCCE) is generally used. It produces a category index of the most likely matching category. It is defined by the next equation 3.49 :

$$SCCE = -\sum_{i=1}^{n} T_i.log(Si) \tag{3.49}$$

Where Si are the Softmax probabilities and Ti the labels.

3.5.2 Machine Learning Methodes

3.5.2.1 Support Vector Machines

Support Vector Machines (SVMs) are a group of supervised learning algorithms used for tasks such as classification, regression, and outlier detection. SVM works by finding the optimal decision boundary, known as a hyperplane, that separates data points from different classes.

An overview of how SVM operates:

Classification: For classification problems, SVM seeks to partition data points into distinct categories by identifying the optimal hyperplane. This hyperplane is selected in such a way that it maximizes the margin, which refers to the distance between the hyperplane and the closest data points from each class.

Mathematically, the decision boundary (hyperplane) in a 2D space can be expressed as:

$$w \cdot x + b = 0 \tag{3.50}$$

where w is the weight vector (perpendicular to the hyperplane), x is the input vector, and b is the bias term.

SVM aims to maximize the margin M between the support vectors and the hyperplane. This margin is defined as:

$$M = \frac{2}{\|w\|} \tag{3.51}$$

To maximize the margin, SVM solves the optimization problem:

$$\min \frac{1}{2}\|w\|^2 \tag{3.52}$$

subject to the constraint that each data point is correctly classified (with a margin):

$$y_i(w \cdot x_i + b) \geq 1 \quad \text{for all } i \tag{3.53}$$

where $y_i \in \{-1, 1\}$ are the labels of the data points.

Support Vectors: The data points closest to the hyperplane are called support vectors. These points are the most informative because they define the margin and the position of the hyperplane. Only the support vectors affect the decision boundary, which makes SVM more efficient since it doesn't need to consider all data points.

Non-linear Separation: When dealing with data that is not linearly separable, SVM can transform the input space into a higher-dimensional feature space. By doing so, it can identify non-linear decision boundaries through an implicit projection of the data into a higher-dimensional space [92].

The figure 3.19 illustrates an example of an SVM :

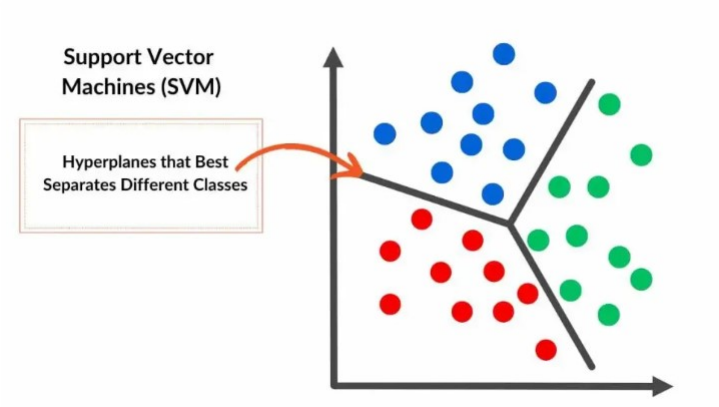


Figure 3.19: SVM example

3.5.2.2 K-Nearest Neighbors

K-Nearest Neighbors (K-NN) is a simple, yet effective, supervised learning algorithm used for both classification and regression tasks. K-NN is based on the idea that data points with similar features are likely to belong to the same class. Rather than making assumptions about the data, K-NN directly compares new examples with stored instances to predict labels or values.

1. **Training Phase:**

   One unique aspect of K-NN is that there is no explicit training phase like in other machine learning algorithms. Instead, K-NN stores the entire training dataset, and predictions are made at runtime by comparing new examples to the stored data.

2. **Distance Metric:**

To classify or predict a new data point, K-NN computes the distance between the new point and all points in the training set. Several distance metrics can be used, but the most common one is Euclidean distance:

$$d(x, x_i) = \sum_{j=1}^{n} (x_j - x_{ij})^2$$

where x is the new data point, x_i is a data point in the training set, and n is the number of features.

3. **Choosing K:**

   K is the number of nearest neighbors the algorithm will consider when making a prediction. The value of K is a critical parameter in K-NN, as it directly influences the outcome. For classification, the majority label among the K nearest neighbors is chosen as the predicted class.

4. **Classification:** In a classification task, the K-NN algorithm works as follows:

   a. For each new data point, calculate the distance to all points in the training dataset.
   b. Sort these distances in ascending order and select the K closest neighbors.
   c. Determine the most frequent class label among these neighbors. This is done by a majority voting system, where the class with the highest frequency among the K neighbors is assigned to the new point.

$$\hat{y} = \text{mode}\,\{y_i \mid x_i \in \text{K nearest neighbors}\}$$

5. **Weighted K-NN:**

   In some cases, neighbors that are closer to the query point are considered more important than distant ones. Weighted K-NN applies weights to the neighbors based on their distance, giving more influence to closer neighbors when making predictions.

6. **Handling Multi-Class Classification:**

   K-NN can handle multi-class classification tasks naturally. Instead of having just two classes, it can assign a label from three or more classes by selecting the majority class from the nearest neighbors.

The Figure 3.20 provides an explanatory illistration of the method :

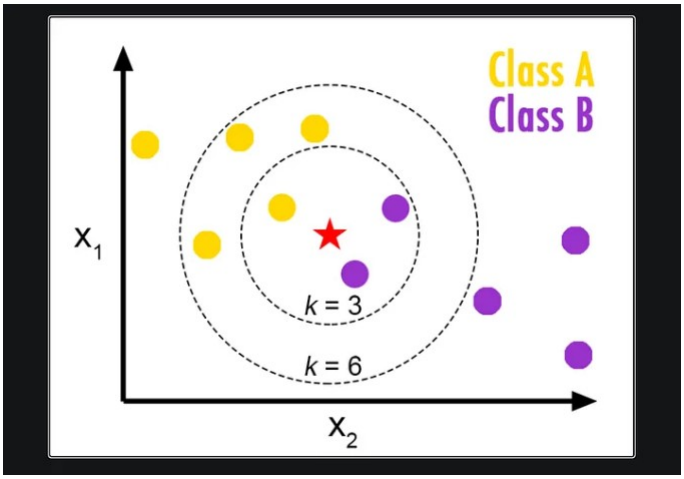


Figure 3.20: k-Nearest Neibghors

3.5.2.3 Random Forest Classifier

Random Forest is a widely used supervised learning algorithm that leverages decision trees and is recognized for its versatility and ease of use. It is particularly effective for addressing classification problems.

The construction of a Random Forest involves creating multiple decision trees, each trained on a distinct subset of observations. When making predictions, the Random Forest combines the predictions from each individual tree by averaging them [32].

The Figure 3.21 provides an explanatory illistration of the method :

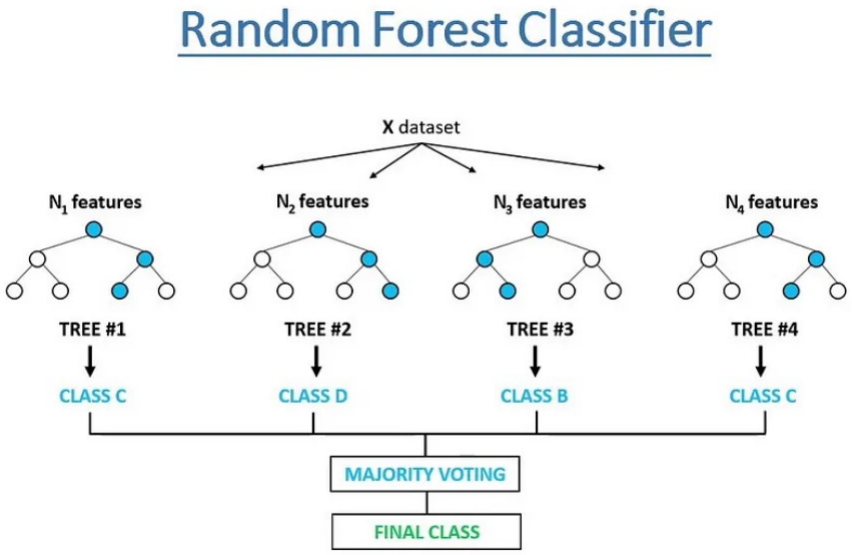


Figure 3.21: Random Forest Classifier

Random Forest offers several advantages :

- It addresses the issue of overfitting by averaging predictions from multiple trees. As a result, Random Forest achieves higher predictive accuracy compared to a single decision tree.

- Moreover, the Random Forest algorithm aids in identifying important features within a dataset, providing valuable insights [32].

However one of the disadvantages like Training a Random Forest can be computationally intensive and time-consuming, especially with a large number of trees and large datasets. This can lead to longer training times compared to simpler models, and sometimes compared to Convolutional Neural Networks.

3.5.3 Accuracy

Accuracy is a common metric used yo quantify the extent to which a model makes correct predictions. The formula for calculating accuracy is provided below 3.54 [92]:

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN} \tag{3.54}$$

Where :

TP, FP: True and False Positives predictions respectively,

TN, FN: True and False Negatives predictions respectively,

3.5.4 Epochs

An epoch represents a complete iteration through all training data during the training process of a machine learning model [92].

3.6 Conclusion

In this chapter, we explored the Human Visual System (HVS) and how its structure inspires advanced techniques for iris recognition. We began with the physiological analysis of the visual pathways, leading to multichannel models and Perceptual Channel Decomposition (PCD), which inspired key feature extraction methods. Specifically, we adopted Discrete Wavelet Transform (DWT) to replicate PCD by capturing both spatial and frequency details, and Gabor filters, which mimic the brain's oriented filters to detect patterns in visual input. In addition, we discussed the importance of preprocessing steps such as image enhancement, reflection removal, segmentation using the Circular Hough Transform, and normalization via Daugman's Rubber Sheet Model, all of which prepare the iris image for feature extraction. For classification, we evaluated SVM, KNN, and Random Forests, identifying their potential in biometric applications. Together, these HVS-based methods form the core framework for our approach, integrating biologically inspired feature extraction and effective preprocessing with machine learning techniques to deliver an optimized iris recognition system.

Chapter 4

Proposed Approach and Results

4.1 Introduction

In this chapter, we aim to provide an in-depth and detailed presentation of our proposed approach. We will systematically cover each component of the system, from the initial phase of iris detection through to the final stage of classification and matching. Additionally, we will thoroughly examine the results obtained, offering detailed explanations and valuable insights throughout the discussion.

4.2 Material used

4.2.1 Programming Langauges and Libraires

4.2.1.1 Python

Python is a high-level, interpreted programming language known for its simplicity, readability, and ease of use. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it versatile for various tasks. Some libraries used in our work :

- **OpenCV :** The huge open-source library for computer vision, machine learning, and image processing also plays a vital role in real-time computer vision systems.
- **Scikit Learn :** Scikit Learn is a Python library developed for Machine Learning. It includes various algorithms for Classification, Regression and Clustering.
- **Tensorflow :** TensorFlow is an open-source end-to-end platform for creating Deep Learning applications. It is a symbolic math library that uses data flow and differentiable programming to perform various tasks focused on training and inference of deep neural networks. It allows developers to create machine learning applications using various tools, libraries, and community resources.
- **Skimage :** scikit-image (or skimage) is a Python library designed for image processing. It provides a collection of algorithms and tools for tasks such as filtering, edge detec-

tion, segmentation, feature extraction, and transformations. Built on top of libraries like NumPy and SciPy.

- **Pywt :** pywt (PyWavelets) is a Python library for wavelet transformations, providing a wide range of tools for discrete wavelet transforms (DWT), continuous wavelet transforms (CWT), and related operations. It is commonly used in signal processing, image compression, and feature extraction, particularly where multi-scale analysis is necessary. pywt offers various wavelet families and supports multi-dimensional data.

4.2.1.2 C/C++

C is a general-purpose, procedural programming language that is well-known for its performance and control over system resources, whime C++ builds upon C by adding object-oriented programming (OOP) features and supporting high-level abstractions, making it suitable for both system and application-level development.

4.2.2 Graphic Card Nvidia Geforce RTX 2060 Super

The NVIDIA GeForce RTX 2060 Super is a mid-range graphics card introduced in 2019 as part of NVIDIA's RTX 20 series. It offers significant performance for gaming, but it is also highly useful in the field of deep learning, particularly for training models like CNNs, that is why it was used for the classification part of our model.

4.2.3 Datasets

4.2.3.1 Datasets Used

To ensure comparability with results from the articles listed in the state of art chapter, we decided to utilize the widely used iris datasets for evaluating our performances. The datasets used are :

- CASIA-Iris V-4 that includes 8 sub datasets [93].
- IIT Delhi Iris Database.

Exploiting the IITD Database :

IITD Database contains iris photos from 224 classes (users). The bitmap (*.bmp) format is used for all of the images. All of the subjects in the dataset are between the ages of 14 and 55, with 176 men and 48 girls. The database contains 2240 photos with a resolution of 320 × 240 pixels. As a result, there are 224 classes in this dataset, each having ten sample iris photos, and all of these images were taken indoors using the near-infrared (NIR) wavelength. Figure 4.1 shows 16-sample images from IITD Iris dataset.

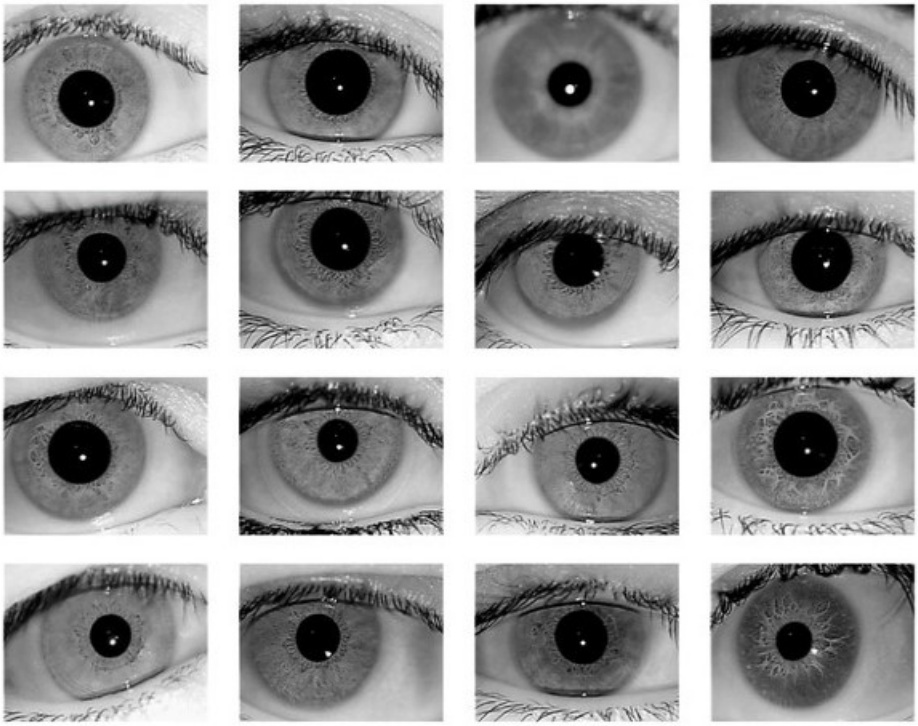


Figure 4.1: A Sample of IITD Database

We preprocessed all the images of each dataset and finally stored them with their respective label (identifier of the person) to feed them to the classifier. The details about the preprocessing phase of the dataset will be seen in the next section.

Some statistics about the used datasets as seen in table 4.1

Datasets	Classes	Images/Class	Images/Dataset
IITD	224	10	2240
Casia V-4	1000	20	20000

Table 4.1: Datasets Statistics

4.2.3.2 Dataset Cleansing

After the iris detection process, certain images may still lack informative content and have the potential to mislead our model during training. Therefore a manual data cleansing is crucial to ensure the optimisation of our model.

4.2.3.3 Data Augmentation

After completing the cleansing process, it is important to ensure that our model is fed with a sufficient number of images. To achieve this, we employ Data Augmentation techniques which help us increase the number of images per class.

4.3 Iris Detection

The preprocessing phase of iris recognition is a crucial step in ensuring accurate and reliable biometric identification. This phase prepares the raw iris image for further processing, such as feature extraction and matching, by enhancing its quality and isolating the relevant iris region. Since iris images can be affected by noise, lighting variations, reflections, and occlusions, preprocessing is designed to address these challenges.

4.3.1 PreProcessing

The preprocessing substage is the crucial part of the iris recognition system, due to reflections, which pose a problem that can highly impact our recognition system, so we proposed a contrast manipulation method to remove the informations added by reflections.

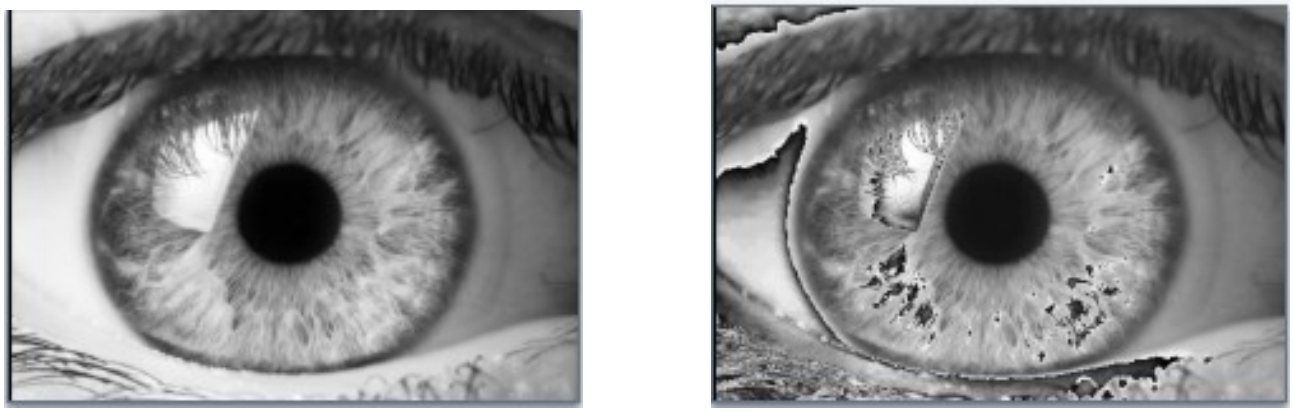
4.3.1.1 Threshold

We are applying binary thresholding with a threshold value of 225 to effectively isolate bright regions in the image, specifically the reflections. By setting this high threshold, we ensure that only the brightest pixels, which correspond to reflections, are highlighted while the rest of the image is suppressed. We then use the resulting binary image as a mask, allowing us to isolate and remove these reflections from the original iris image, ensuring that they don't interfere with subsequent processing steps like segmentation and feature extraction.

4.3.1.2 Histogram Equalization

The second step of our preprocessing consists of histogram equalization to adjust the brightness of the isolated reflections in the image. This technique redistributes the pixel intensities across the image, helping to balance the brightness levels and enhance any hidden textures within the reflections. By applying this method, we ensure that the details within these bright regions are more visible, which improves the accuracy of later stages.

Here is an example that shows the difference between an original image and the image with reduced reflections :



(a) Original image (b) Reduced Reflections

Figure 4.2: Comparison between an original iris image and preprocessed iris image

The results show that our approach effectively removes reflections while preserving the entire iris region, including areas affected by glare. By isolating and adjusting the reflections, the underlying iris texture remains intact, ensuring reliable feature extraction for more accurate iris recognition.

4.3.2 Edge Detection

To identify the iris and associate it with an individual, we must first determine its location. This process begins by detecting the outer and inner edges of the iris within a complete image of the eye.

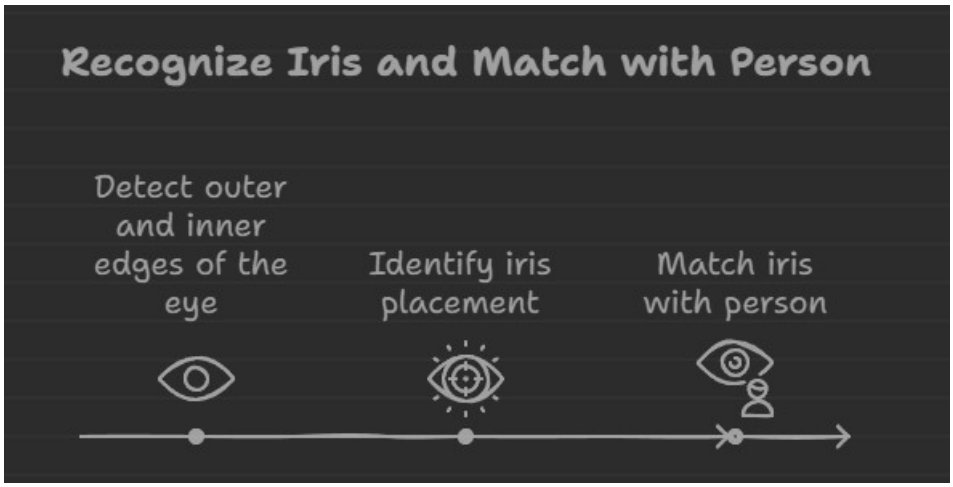


Figure 4.3: Iris Inner and Outer Edges

To achieve this, we used the Canny edge detector to accurately identify the edges within the iris image. The Canny filter operates in multiple steps to enhance accuracy, starting with Gaussian filtering to smooth the image and reduce noise. This prevents false edges from being detected due to minor fluctuations in pixel intensity.

Next, the gradient calculation determines areas of sharp intensity change, which are likely to represent edges.

The non-maximum suppression step then refines this by ensuring that only the most prominent edges are retained, effectively sharpening the image.

Finally, double thresholding is applied to classify edges based on intensity, followed by edge tracking by hysteresis to connect the strongest edges and discard weak, irrelevant ones.

Another approach we used to test the edge detection of iris is the Laplacien of Gaussian; however, the Canny edge detector tends to outperform the Laplacian of Gaussian because of its Superior noise reduction capabilities and More efficient edge refinement and extraction, specifically targeting strong, relevant edges (such as the boundaries of the iris).

By using the Canny filter, we ensure that only the most significant edges around the iris are detected, which aids in accurate segmentation for later stages of recognition. In the edge detection output image, the boundaries of the iris are clearly defined, highlighting how this method effectively isolates the iris from other regions, improving the precision of feature extraction in iris recognition.

4.3.3 Iris Localization and Isolation

In the segmentation stage of iris recognition, we employ the Circular Hough Transform (CHT) to accurately detect the inner and outer boundaries of the iris. The CHT is particularly effective for this task because it identifies circular shapes in an image, allowing us to pinpoint the edges of the iris with precision. The steps of the CHT as seen in the third chapter :

- **Probable Circle Detection :** The first step of the CHT is to identify all potential circles in the image based on edge detection results. This involves mapping points in the image space to a parameter space where each point corresponds to a possible circle defined by its center coordinates and radius.
- **Filtering Process :** After identifying the probable circles, a filtering process is applied to eliminate any incorrectly detected circular shapes. This step focuses on the specific range of the maximum and minimum iris radius, ensuring that only circles that match the expected size of the iris are considered.
- **Edge Selection :** Following filtering, the CHT selects the most highly linked edges that correlate with the detected circles. This selection helps refine the detection, leading to the recognition of a rough iris boundary that outlines both the outer and inner contours of the iris.

The results of the Hough Transform used for the iris localization is shown in figure 4.4

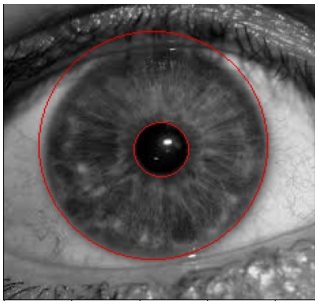


Figure 4.4: Circles of Iris using Hough Transform

Once the inner and outer circles are established using the CHT, we create two binary masks: Mask A, representing the outer circle, and Mask B, representing the inner circle. To isolate the iris region effectively, we multiply these masks with the original image. This multiplication results in a new image that retains only the iris area, while suppressing all other regions. The result is shown in figure :

Figure 4.5: The obtained Iris after Segmentation

Compared to the U-Net method previously employed for a test, the Hough Transform approach proves to be more efficient due to its targeted detection of circular shapes, which minimizes computational complexity and enhances accuracy. In the resulting segmentation image, the clear delineation of the iris and pupil boundaries is evident, illustrating the effectiveness of the CHT and the binary mask method in achieving precise iris segmentation for further processing.

4.3.4 Normalization

In the normalization step of iris recognition, we transform the circular iris region into a rectangular format using Daugman's Rubber Sheet Model, which ensures that the iris is represented in a consistent, comparable format. This transformation is necessary because the iris is a circular object, and its features can be distorted due to factors like pupil dilation, making direct comparison between images challenging. Daugman's model addresses these issues by converting the polar coordinates of the iris into a rectangular grid.

The steps of daugman's rubber sheet model as seen in the third chapter :

- Polar-to-Rectangular Transformation: In Daugman's Rubber Sheet Model, each point on the iris, initially defined in polar coordinates (r, θ), is mapped to a corresponding point in a rectangular coordinate system. This transformation converts the varying radial distances from the center of the pupil to the outer iris boundary into fixed-length segments, ensuring that the entire iris region is captured evenly.
- Uniform Sampling : To create this mapping, the model samples points radially from the pupil boundary to the outer iris boundary for each angular position (θ) around the iris. This approach ensures that both the inner and outer boundaries of the iris are preserved and accounted for, creating a uniform grid for feature extraction.
- Size Standardization : Once the iris has been mapped from polar to rectangular coordinates, the resulting image is resized to a standard dimension, typically 128x128 pixels.

This size standardization is important for ensuring that the feature extraction process is consistent across different iris images, regardless of variations in iris size, dilation, or camera perspective.

The use of Daugman's Rubber Sheet Model offers several advantages. By normalizing the iris in this manner, we eliminate variations caused by pupil dilation, head tilt, or slight distortions in the original image. This creates a more reliable and consistent representation of the iris for comparison

Figure 4.6: A Normalized Iris

In the final result as we see in figure 4.6, we obtain a 128x128 pixel rectangular image, where the iris texture is evenly spread and normalized, ready for further feature extraction.

4.3.5 Enhancement

In the next step of iris recognition, we focus on enhancing the quality of the image to improve the accuracy of subsequent feature extraction. This method have been used by my colleague Maroua Rekrouk in her thesis[94] and have been proven to be so efficient. To achieve this enhancing, we adjust the brightness and contrast of the normalized iris image through a series of methods, each contributing to the enhancement process. The steps involved are as proposed in [95] :

1. **Median Filter :** The first enhancement step involves applying a median filter to the image. This filter is particularly effective for reducing salt-and-pepper noise (random white and black pixels) that can affect the clarity of the image. The median filter works by replacing each pixel's value with the median value of the neighboring pixels within a defined window. This smooths out any abrupt variations in pixel intensity caused by noise while preserving the edges of the image, which is crucial for maintaining the sharp boundaries of the iris texture. By using this filter first, we ensure that the image is noise-free before moving on to further enhancements.

2. **Histogram Equalization :** After denoising the image, we perform histogram equalization to improve the contrast. This method redistributes the pixel intensities across the entire image to achieve a more uniform brightness distribution. It works by stretching the range of intensities so that the dark and light areas are balanced, making the finer details of the iris texture more visible. This is particularly useful for enhancing hidden features that may not have been apparent due to poor contrast, improving the image's overall quality for feature extraction.

3. **Top-Hat and Bottom-Hat Filters :** The final enhancement step involves applying top-hat and bottom-hat filters, which are morphological operations used to enhance bright and dark regions, respectively. The top-hat filter is designed to highlight the bright features of the image, emphasizing regions that are brighter than their surrounding areas. This is particularly useful for bringing out finer details and enhancing any bright textures within the iris. The bottom-hat filter, on the other hand, enhances dark features by emphasizing areas that are darker than their surroundings. By applying both filters, we improve the visibility of both light and dark textures within the iris, ensuring that all relevant details are highlighted. This combined approach provides a more balanced and detailed representation of the iris texture, making it more suitable for subsequent analysis.

In the enhanced image, after applying the median filter, histogram equalization, and top/bottom-hat filters, the quality of the iris texture is significantly improved. Noise is reduced, contrast is balanced, and both bright and dark features are highlighted, making the iris details much clearer. This enhanced image will provide a stronger foundation for accurate feature extraction, ultimately improving the reliability of the iris recognition system.

4.3.6 Dataset Enhancing

To improve the performance and reliability of our iris recognition system, we propose specific methods for cleansing the dataset and performing data augmentation, ensuring that the images are of high quality and diverse enough for robust model training.

4.3.6.1 Dataset Cleansing

To start, we implemented a systematic image quality assessment to filter out any poor-quality iris images that could affect recognition performance. For this, we proposed using :

- **Blur detection :** By applying a variance of the Laplacian filter, we assess the sharpness of each image. Any excessively blurry images that obscure iris details are removed to ensure clarity in the dataset.
- **Reflection detection :** Since reflections can degrade image quality, we used the reflection isolation method from the preprocessing phase to identify images with large and unrecoverable reflection regions. These images are excluded to maintain high-quality data for feature extraction.

Additionally, we ensure that segmentation is consistently accurate by verifying the output of the Circular Hough Transform (CHT). Images where the CHT fails to correctly detect the inner and outer iris boundaries are flagged and removed from the dataset. This ensures that only well-segmented images proceed to the next stages.

4.3.6.2 Data Augmentation

After Cleansing our datasets, some of the classes became unbalanced due to the removal of non desired images. Therefore, in order to balance the dataset as well as making our model

adaptable to varying conditions, as well as having an important amount of data to be fed to the model. We applied some data augmentation techniques :

- To account for minor rotations or slight misalignments during image capture, we introduced small rotations and translations to the iris images. This helps the recognition system become more resilient to small changes in the camera angle or user positioning.
- Brightness adjustments: We proposed adjusting the brightness of the images to simulate different lighting conditions, which improves the system's ability to handle images captured in both low and high lighting environments.

this process resulted in a dataset containing the same number of images per class (20), addressing the initial issue of data imbalance.

The results of the data augmentation, including the details of the number of images per dataset and the total are in table 4.2:

Datasets	Number of Classes	Images/Classe	Images/Dataset
IITD	224	40	8960
Casia v-4	1000	40	40000
		Total	48960

Table 4.2: Our datasets after proceeding data augmentation

4.4 Features Extraction

Now in the most crucial part of our model, the feature extraction part plays a pivotal role as it forms the foundation for distinguishing one iris from another based on its unique patterns. This step is crucial because it translates the raw, preprocessed image into a set of identifiable features that can be efficiently compared across different images.

In fact, feature extraction is central to the study of Human Visual System (HVS) based models, as these models aim to mimic the way humans perceive and process visual information. By carefully designing feature extraction methods, we can enhance the system's ability to capture the most relevant information from the iris, mirroring the HVS's ability to process complex visual stimuli.

In our approach, we focus on two feature extraction techniques that closely align with the Human Visual System Modelization of Perceptual Channel Decomposition: Discrete Wavelet Transforms (DWT) and Gabor Filters. These methods are well-suited for iris recognition because they decompose the image into various frequency and orientation components, mimicking how the human eye and brain process visual information at different scales and orientations. These methods provide a multi-resolution analysis that allows for the capture of both fine and coarse details of the iris texture.

We opted for Discrete Wavelet Transforms and Gabor Filters because they are the most effective at simulating this perceptual decomposition, breaking down the iris image into key frequency components that resemble the way the HVS processes visual stimuli. While the DCP method could theoretically offer more precise modeling of human visual perception, we decided

against using it in our system. The reason lies in the complexity of implementing DCP filters, particularly the absence of well-defined formulas for calculating the required filter frequencies. This lack of established equations presents practical difficulties in designing and implementing DCP-based feature extraction.

In order to know how efficient our HVS based models (DWT and Gabor Filters) are, we decided to also use 2 non HVS based methods : Discrete Cosine transform consisting of only Frequency domain and Histogram of Gradients who operates purely in the spatial domain by analyzing gradients.

4.4.1 Discrete Wavelete Transforms (DWT)

In our model, we integrated the Discrete Wavelet Transform (DWT) which is a powerful tool that breaks down an image into different frequency components, allowing us to analyze both global and fine details of the iris. Its multi-scale approach resembles perceptual channel decomposition, a method used in human visual system modeling, which captures how humans process visual information across different spatial frequencies. This similarity motivated us to select DWT for our iris recognition system.

We chose DWT for its ability to represent the intricate textures of the iris at various resolutions. The decomposition process captures both low-frequency information (overall shape and structure) and high-frequency details (fine textures and edges). This multi-level analysis is essential for effectively identifying the unique features of each iris, which form the basis of our recognition system.

4.4.1.0.1 The Decomposition Process

1. **Choose the Wavelete :** We chose 4 families of Wavelets to test our Model :
    a. **Haar Wavelet :** The Haar wavelet is the simplest and oldest wavelet, first introduced by Alfréd Haar in 1909. It is a step function that oscillates between two values (1 and -1) and is non-continuous. The Haar wavelet focuses on detecting sharp transitions in the image, which makes it particularly useful for edge detection. It is a very simple and efficient wavelet, but it lacks the ability to represent smooth or detailed variations in images.
    b. **Daubechies Wavelets :** The Daubechies wavelets, introduced by Ingrid Daubechies, are compactly supported orthogonal wavelets that are designed to represent smoothness and regularity. They have a series of wavelets indexed by their order (e.g., Daubechies 9, 11), with higher-order Daubechies wavelets being longer and better at capturing fine details in irregular structures. These wavelets are particularly well-suited for applications like image and signal processing due to their ability to represent complex textures with compact support.
    c. **Symlet Wavelets :** The Symlet (Symmlet) wavelets are a variation of Daubechies wavelets that aim to have better symmetry properties. They are more symmetrical than Daubechies wavelets, which improves their performance in applications where symmetry is important. Symlet wavelets still have compact support and orthogonality, making them useful for signal processing and image decomposition. The higher-order Symlets (e.g., Symlet 5, 6, 8) provide better localization in both time and frequency domains, making them ideal for capturing fine details in images.

d. **Biorthogonal Wavelets :** Biorthogonal wavelets differ from orthogonal wavelets in that they use two sets of filters: one for decomposition and another for reconstruction. This allows for perfect reconstruction with smoother wavelets and greater flexibility in design. The Biorthogonal 3.5 wavelet, for example, is designed to provide good symmetry and smoothness, which makes it well-suited for applications requiring precise reconstruction, such as image processing. This property makes biorthogonal wavelets useful for capturing intricate details while ensuring minimal loss during reconstruction.

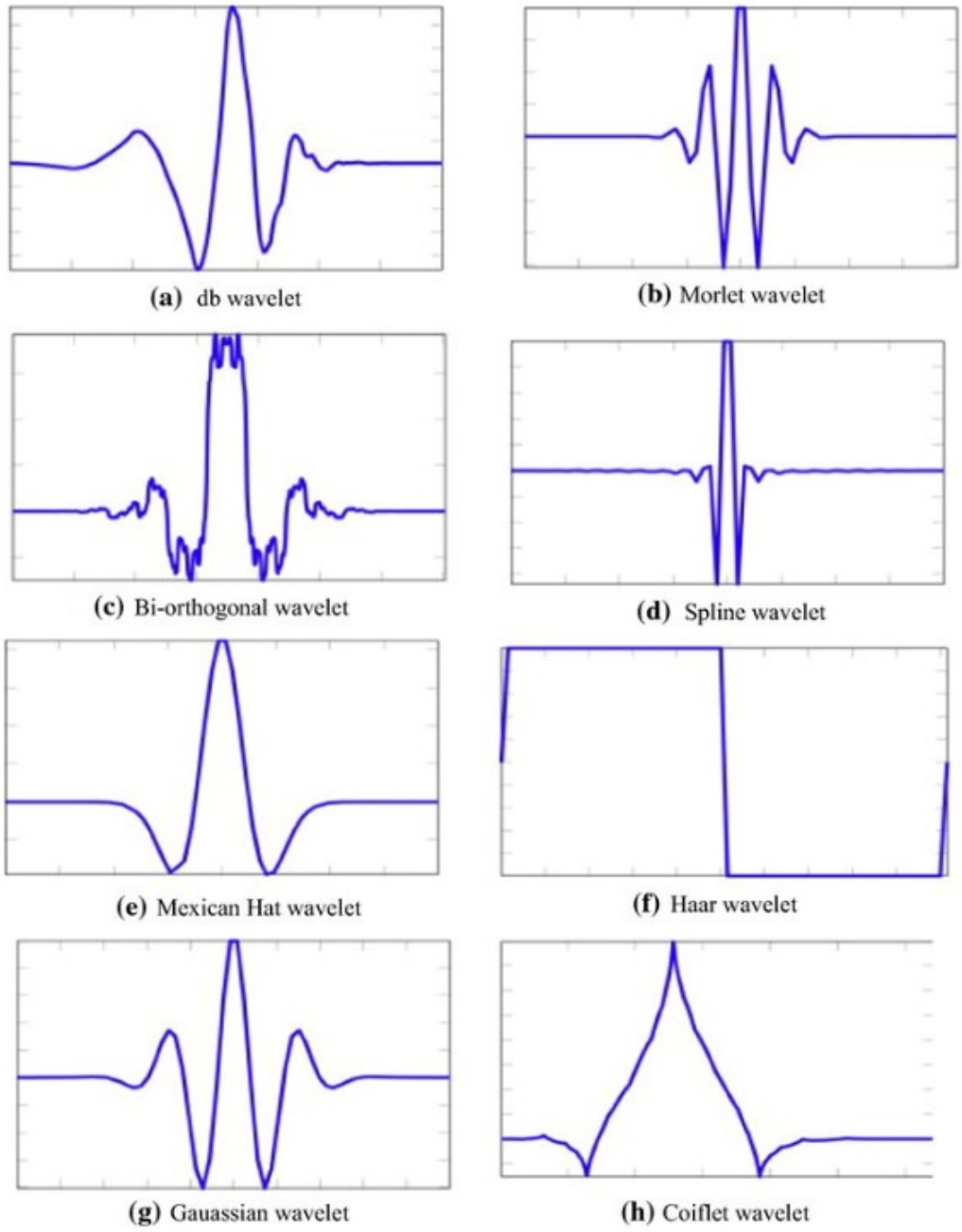


Figure 4.7: Wavelets Families

2. **Low-Pass and High-Pass Filters :**

   The low-pass filter extracts the smooth, global structure of the image (large-scale details), while the high-pass filter captures the sharp transitions and edges (fine details).

For each wavelet family, the filters are derived from the wavelet's mathematical properties. These filters are convolved with the image to separate the different frequency components.

3. **Downsampling :**

   After applying the low-pass and high-pass filters, we downsample the image by a factor of 2 in both the horizontal and vertical directions. This reduces the image size by half, focusing on the most important information while discarding unnecessary data (high-frequency noise).

   The output is a set of four sub-images:
   -LL (Low-Low): Approximation sub-band, containing low-frequency components in both directions (captures overall structure).
   -LH (Low-High): Contains low-frequency in the horizontal and high-frequency in the vertical direction.
   -HL (High-Low): Contains high-frequency in the horizontal and low-frequency in the vertical direction.
   -HH (High-High): Captures high-frequency components in both directions (fine details like edges). Figure 4.8 represents the LL, LH, HL and HH sub-bands outputs by applying the DWT on our normalized iris.

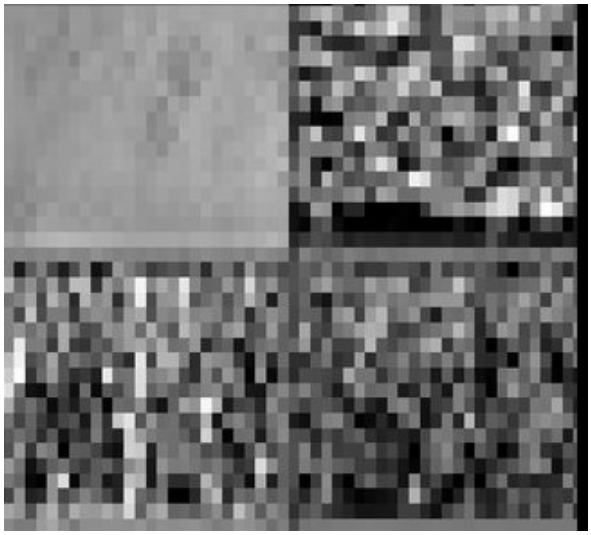


Figure 4.8: Applying DWT on a normalized Iris

4. **Recursive Application :**

   We recursively apply the same process (low-pass and high-pass filtering, followed by downsampling) to the LL sub-band, which contains the approximation (or coarse) representation of the image. This creates a multi-level decomposition, where each level captures more detailed features of the iris.

   For Haar wavelet, we applied only one level of decomposition due to its simplicity and focus on sharp transitions. For Biorthogonal 3.5 and Daubechies 9/11, we performed four levels of decomposition because these wavelets can handle more complex structures in the iris. Symlet 6/8 was decomposed into five levels due to its ability to capture even finer details, but we stopped at six to avoid losing important global information and avoiding the inclusion of too much high-frequency noise. While deeper decomposition may capture additional details, it can also amplify noise and obscure the important global features of the iris. Fiver levels allowed us to extract relevant patterns while maintaining computational efficiency and accuracy in recognition.

4.4.2 Gabor Filters

The Gabor filter is a classical method widely utilized for feature extraction and texture analysis. Given that gabor's filters ability to mimic how the human visual system processes images, particularly through the decomposition of images into multiple spatial frequencies and orientations, similar to how the visual cortex in the brain interprets visual stimuli, and on capturing the texture of the iris, the Gabor filter emerged as an intriguing choice.

The Gabor filter combines the concepts of frequency and orientation selectivity, making it particularly effective in capturing textural details in an image. It uses a set of sinusoidal functions modulated by a Gaussian window to analyze an image at different frequencies and orientations. As seen in the 3^{rd} chapter, by varying the following parameters : Wavelength (λ), Orientation (θ), the standard deviation of the Gaussian (σ), Phase offset (ψ) and the Aspect ratio (γ), we obtain a set of gabor filter Bank

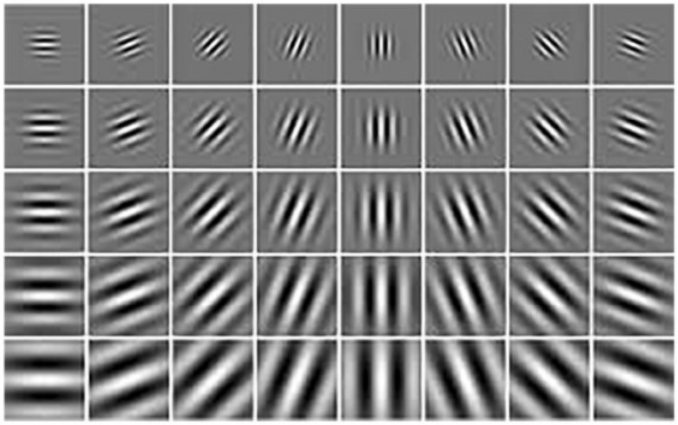


Figure 4.9: Gabor filters for 8 orientations and 5 wavelengths

Once the Gabor filter bank was defined with these parameter variations, we convolved each filter with the iris image. This convolution process allowed us to extract multi-scale, multi-orientation texture information from the iris, producing a series of feature maps. These feature maps represent different aspects of the iris's intricate texture and provide detailed information on the spatial frequency content at various orientations. The Figure 4.10 illustrates the output of Gabor filter Bank applied on a IITD's image :

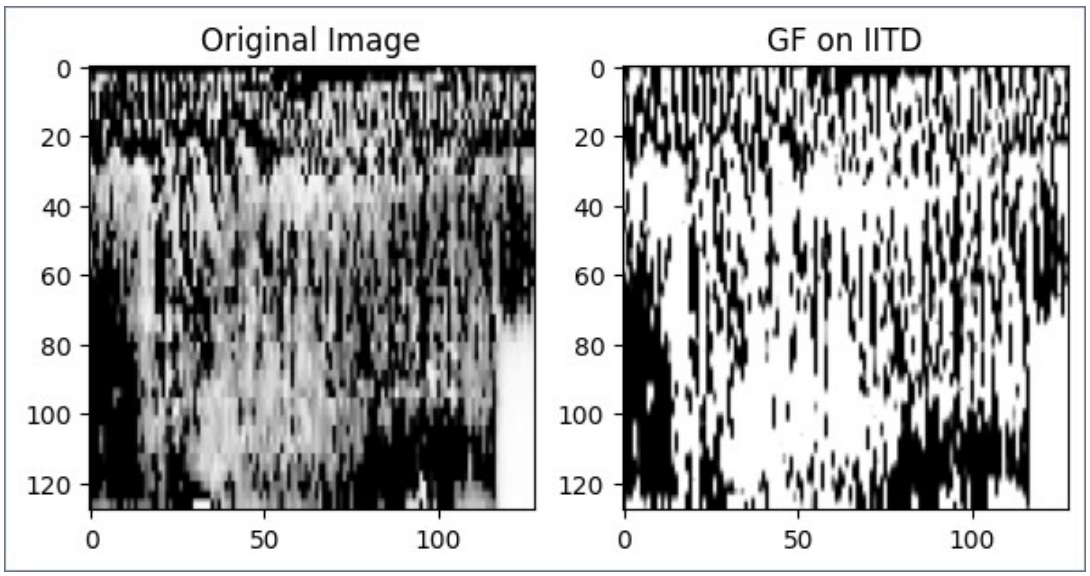


Figure 4.10: Gabor Filters used on an IIFD Database image

By implementing Gabor filters in our system, we not only ensure that we are capturing the most critical details from the iris but also mirror the biological processes that naturally occur in the brain when analyzing visual information. The result is a highly effective feature extraction technique that captures the complex textures of the iris, making it a powerful tool for accurate and reliable iris recognition. Also, its ability to extract relevant features from iris images makes it a valuable tool in our pursuit of accurately representing and distinguishing different iris textures.

4.4.3 Discrete Cosine Transform

DCT is a powerful technique used for converting spatial domain data into the frequency domain, which allows us to analyze the frequency components of an image. We applied the 2D DCT to the preprocessed iris image. This transformation decomposes the image into a sum of cosine functions oscillating at different frequencies. After transforming the image, we extracted a specific number of the lowest-frequency DCT coefficients. These coefficients represent the most significant components of the iris image and contain the essential information needed for recognition.

4.4.4 Histogram of Oriented Gradients

HOG features are particularly useful for iris recognition because they emphasize the shapes and structures present in the iris texture. By focusing on the orientation and distribution of gradients, HOG effectively captures the distinct features of the iris, such as ridges and furrows, which are essential for accurate identification. The robustness of HOG against variations in lighting and occlusion further enhances its applicability in real-world scenarios.

4.4.5 Features Extraction Results

The table 4.3 contains the number of features collected from each method, although the number of features doesn't really show how good is a technique, but knowing how much features extracted from each method could give us the big picture of how each method works and so its results.

Method		**Number of Features**
DWT	One level of Decomposition	16000
	4 Levels of Decomposition	21760
	5 Levels of Decomposition	21824
Gabor Filters		365000
DCT		1024
HOG		8100

Table 4.3: The number of features extracted for every method

4.5 Feature Selection

In the feature selection phase of our iris recognition system, we employ Principal Component Analysis (PCA) to reduce the dimensionality of the extracted features. Feature selection is a crucial step because it helps in focusing on the most important and discriminative features, while discarding redundant or irrelevant information. This not only enhances the system's accuracy but also improves computational efficiency, which is essential for real-time application.

After extracting the iris features using Discrete Wavelet Transforms (DWT) and Gabor Filters, and even in the convolutional Neural Networs which will be seen in the next section, the resulting feature set can be quite large, especially since these methods capture a wide range of frequency and orientation information. Directly using all these features for classification would lead to overfitting, increased computational cost, and potential redundancy. This is where PCA comes into play, as it allows us to reduce the dimensionality while retaining the most significant information.

4.6 Classification

4.6.1 Machine Learning Based Methods

Due to the time, energy, and resource-intensive nature of DNN classifiers, it was deemed necessary to utilize alternative classifiers, such as SVM and Random Forest, which are comparatively less resource-consuming.

4.6.1.1 Support Vector Machine

The objective of this algorithm is to find a hyperplane in an n-dimensional space that separates the data points to their potential classes. The hyperplane should be positioned with the maximum distance to the data points. The data points with the minimum distance to the hyperplane are called Support Vectors.

The objective is to correctly classify as many data point as possible by maximizing the margin from the Support Vectors to the hyperplane while minimizing the term $w^T w$ means finding the optimal w and b that most samples are predicted correctly.

In our case, we have a multi-class classification. SVM was initially designed for binary classification, but it can be extended to multi-class classification using the One-vs-One (OvO) technique : Train an SVM for every pair of classes.

The number of classifiers necessary for one-vs-one multi-class classification can be retrieved with the following formula 4.1:

$$\frac{n * (n-1)}{2} \tag{4.1}$$

Where n is the number of classes.

In the one-vs-one approach, each classifier separates points of two different classes and comprising all one-vs-one classifiers leads to a multi-class classifier.

4.6.1.2 K-Nearest Neighbors

For the KNN classifier, we first determined several key parameters:

Choosing the value of K : After experimenting with various values, we selected *K=5*. This value was found to provide a good balance between bias and variance, enabling the model to generalize well while minimizing the influence of noise from individual samples.
Distance Metric : We opted to use the Euclidean distance metric for measuring the similarity between feature vectors. The Euclidean distance is effective for capturing the spatial relationship between data points in the feature space, making it suitable for the high-dimensional features extracted from the iris images.

4.6.1.3 Random Forest

Random Forest algorithm is an ensemble learning method that operates by constructing multiple decision trees during training. Here's how we implemented it:

Selecting Hyperparameters : We configured the Random Forest with the following key parameters:

Number of Trees: We chose to create *n-estimators*=100 trees in the forest. This number was sufficient to achieve a good trade-off between computational efficiency and model accuracy.
Maximum Depth: We set the maximum depth of each tree to prevent overfitting, ensuring that the model maintains generalization across unseen data. Random Subset of Features: At each split in the tree, a random subset of features was considered. This randomness promotes diversity among the trees and improves the robustness of the model. Training Phase: The Random

Forest was trained on the feature vectors extracted from the iris images. Each decision tree was built using a bootstrap sample of the training data, which adds an element of randomness to the training process.

Prediction Phase : For predicting the identity of a test sample, each tree in the forest made an independent classification based on its learned structure. The final prediction was made by taking the majority vote from all the trees, leading to a more accurate and stable classification outcome.

The diversity of opinions and perspectives among the trees plays a crucial role in minimizing biases and errors. Consequently, the Random Forest becomes a more reliable and effective model, benefiting from the combined knowledge and insights of its constituent trees [92].

Here is a explanatory example of the method (Figure 4.11):

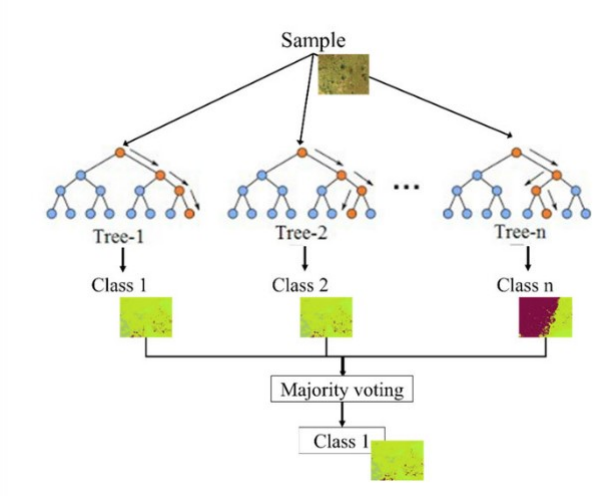


Figure 4.11: Random Forest Algorithm

4.6.2 Deep Learning Classification

Unlike the machine learning classification methods, the CNN not only do the classifications with their dense layers, but extract its own features from whatever features we gave it or the whole image.

The Proposed CNN Architecture :

- **Input Layer:**
  - The input layer accepts iris images that have been preprocessed and normalized, typically sized at 128×128 pixels. This consistent size ensures that the network processes images uniformly.
- **Convolutional Layers:**
  - **Layer 1:**
    - 32 filters of size 3×3, with a stride of 1 and padding set to 'same' to maintain the spatial dimensions. This layer learns low-level features like edges and textures.

        * **Activation Function:** ReLU (Rectified Linear Unit) is applied to introduce non-linearity.
        * **Batch Normalization:** This is implemented after the activation to normalize the output, improving convergence speed and stability.
    - **Layer 2:**
        * 64 filters of size 3×3 with the same padding. This layer captures more complex features by building on the low-level features extracted in the first layer.
        * **Activation Function:** ReLU.
        * **Batch Normalization:** Again applied for improved training.

- **Pooling Layer:**

    - Max Pooling Layer: A 2×2 pooling layer follows the second convolutional layer to down-sample the feature maps, reducing their dimensions while retaining essential information and reducing computational complexity.

- **Flatten Layer:**

    - This layer flattens the pooled feature maps into a one-dimensional vector, which serves as input to the fully connected layers.

- **Fully Connected Layers:**

    - **Layer 1:** A dense layer with 128 neurons, employing the ReLU activation function to learn non-linear combinations of the features extracted by the convolutional layers.
    - **Layer 2:** A dense layer with 64 neurons, again using the ReLU activation.

- **Output Layer:**

    - A final dense layer with softmax activation, producing a probability distribution over the classes (i.e., recognized iris patterns). The number of neurons in this layer corresponds to the number of unique classes in the dataset.

Parameters and Hyperparameters Used

- **Batch Size:** We selected a batch size of 32 during training, which balances memory usage and convergence speed.

- **Epochs:** The model was trained for 30 epochs, ensuring adequate learning without overfitting.

- **Learning Rate:** We used an initial learning rate of 0.001 with the Adam optimizer. The Adam optimizer was chosen for its adaptive learning rate properties, which help in faster convergence.

- **Loss Function:** The categorical cross-entropy loss function was utilized, as it is suitable for multi-class classification problems.

- **Regularization:** Dropout layers with a dropout rate of 0.5 were applied after the fully connected layers to prevent overfitting by randomly setting half of the neurons to zero during training.

4.6.3 Results

4.6.3.1 Table of Results

The results we got when testing the different combination of features extraction/classification methods are shown in the table 4.4

Feature Extraction/Classification	SVM	KNN-5	Random Forest	DNN
DWT (Symlet)	97.2%	94.6%	95.8%	98.5%
DWT (Debauchies)	95.3%	89.9%	94.5%	96.2%
DWT (Haar)	88.7%	88.7%	86.6%	90.3%
DWT (Biorthogonal)	94.7%	89.9%	94%	95.5%
Gabor Filters	94.3%	89.9%	92.8%	95%
DCT	78%	66.2%	76.5%	85.1%
HOG	86.7%	85.2%	85%	88%
CNN	/	/	/	97%

Table 4.4: Test Accuracies of the different combinations of feature extraction x Classification method

4.6.3.2 Discussion

• In our analysis, we found that the Discrete Wavelet Transform (DWT) outperformed all other feature extraction methods, delivering the highest accuracy. This can be attributed to its alignment with the human visual system (HVS). The decomposition of images into different frequency channels using DWT allowed us to retain critical information, both in low and high frequencies. The LL filters captured smooth details and textures, while the LH, HL, and HH filters preserved high-frequency details, such as edges, without being excessively affected by noise. This is likely because DWT effectively manages noise by localizing frequency variations, allowing the system to focus on meaningful signal patterns.

• Among the wavelet families, Symlet and Debauchies provided the best results. Symlet's near-symmetrical structure allowed for better reconstruction of features, while Debauchies' compact support made it suitable for capturing both smooth textures and sharp changes. These wavelets align well with the HVS by mimicking the perceptual channel decomposition, which likely contributed to their superior performance.

• Gabor filters also demonstrated relatively good results, as their multi-scale, multi-orientation filtering mechanism captures texture and frequency information, similar to DWT. However, Gabor's performance was slightly lower due to its sensitivity to parameter tuning, particularly with varying spectral characteristics in the dataset. The fixed parameters may have led to suboptimal results as they could not adapt to the diverse texture patterns present in different iris images.

• The Histogram of Oriented Gradients (HOG) method performed poorly compared to other methods. While HOG is known for its effectiveness in facial recognition, it struggled in iris recognition due to the absence of frequency and time resolution, which is critical for detailed

texture analysis in iris images. Its edge-based feature extraction did not fully capture the rich, fine-grained details of the iris, which are crucial for high accuracy.

• Lastly, Discrete Cosine Transform (DCT) exhibited the worst performance in our experiments. While DCT is widely used for image compression, it lacks the necessary adaptability for feature extraction in iris recognition. DCT is not an HVS-based model and primarily focuses on frequency domain information, without capturing the temporal or spatial changes efficiently, making it less suitable for iris texture analysis.

• When comparing the classification methods, Convolutional Neural Networks (CNNs) stood out as the most powerful in terms of accuracy. The hierarchical structure of CNNs, with their ability to automatically learn spatial hierarchies of features, made them particularly well-suited for handling the complex patterns found in iris images. With each layer in the CNN model progressively learning higher-level features (from simple edges to more complex textures), it efficiently captured the fine details and intricacies of the iris texture. This performance was enhanced by using our proposed architecture, specifically designed for this task, with parameters carefully tuned to avoid overfitting, resulting in consistent accuracy across the datasets.

On the other hand, Random Forest and Support Vector Machines (SVM) also delivered strong results when combined with the high-quality features extracted by methods such as DWT and Gabor filters. Random Forest, with its ensemble learning approach, was robust to noise and outliers, making it effective for our task. The SVM classifier, leveraging its ability to create decision boundaries in high-dimensional spaces, performed well with features that were well-separated and distinctive, like those obtained from wavelet decompositions. However, while these classifiers performed admirably, K-Nearest Neighbors (K-NN) lagged behind. K-NN's sensitivity to noise and its reliance on the distribution of data points made it less reliable for iris recognition, particularly when the data contained slight variations or imperfections. That said, classification was not the primary focus of our study—our emphasis was on feature extraction, where the choice of the right method significantly impacted the model's performance.

• Ultimately, the key takeaway from our study is the clear advantage of human visual system (HVS)-based feature extraction methods, particularly DWT, which was a standout. The decomposition into different frequency channels, along with the ability to capture both fine details and broader patterns, made DWT particularly effective for iris recognition. By selecting Symlet and Debauchies wavelets, which closely mimic perceptual channel decomposition (how the human eye processes images), we were able to achieve superior results. These wavelets balance the need for capturing both sharp edges and smooth textures, closely reflecting how the human visual system interprets natural images. Symlet's symmetry provided more accurate reconstructions, while Debauchies' compact support effectively localized the important features.

In comparison, Gabor filters, while still useful, fell short of DWT because they lacked the adaptability and multi-resolution nature of wavelets. Although Gabor filters are inspired by the receptive fields in the visual cortex and can capture orientations and scales effectively, their performance was more limited due to the need for precise parameter tuning, which can be challenging given the variability of iris textures across individuals. Additionally, HOG and DCT showed the weakest results. HOG's focus on edge detection without frequency analysis made it inadequate for capturing the rich, complex textures of the iris, and DCT, while excellent for compression, failed to capture the multi-level details needed for high-precision recognition, as it doesn't align with how the HVS processes image textures.

In conclusion, our results clearly demonstrate that HVS-based methods, like DWT, are best suited for the challenging task of iris recognition, offering a robust framework for capturing the

intricate textures and details essential for high accuracy.

4.7 Implementation

To ensure that our system performs efficiently, we deployed our models across three distinct hardware platforms: Nvidia Geforce RTX 2060 Super (GPU), AMD Ryzen 5 4650G (CPU), and PYNQ Z2 FPGA. Each platform offers unique strengths in terms of processing power, energy consumption, and latency, which are crucial for real-time applications like iris recognition.

After training the models and testing their accuracy on test and validation sets, we saved the models in the appropriate formats so that it can be used later :
TensorFlow models were saved using **.h5** or SavedModel format.
Scikit-Learn models (SVM, KNN, Random Forest) were saved as **.joblib**.

4.7.1 GPU Implementation

Before implementing our model in the GPU, the following necessary libraries have been installed :

- **Cuda (Compute Unified Device Architecture)** :
  Cuda is a parallel computing platform and programming model developed by NVIDIA. It enables developers to use graphics processing units (GPUs) for general-purpose computing (GPGPU). In other words, Cuda allows GPUs to solve complex problems more efficiently than CPUs for certain tasks, especially those requiring a large number of parallel computations, which is often the case in deep learning.
- **cuDNN (CUDA Deep Neural Network library)** : cuDNN is a GPU-accelerated library for deep neural networks. It provides highly optimized routines for standard operations such as convolutions, activations, and normalization. cuDNN is designed to be easily integrated into high-level frameworks like TensorFlow, simplifying development and improving the performance of deep learning applications.
- **CuPY :**CuPy is a python library which mimics NumPy arrays but operates on the GPU. Given that the scikit-learn models (like SVM and KNN) doesn't natively support GPU execution, we use CuPy to transfer data to the GPU.

Using the CUDA two libraries, TensorFlow will automatically detect the GPU and perform computations accordingly, greatly facilitating the development and training of large-scale deep learning models. And using the CuPy library we can implement the machine learning models on the GPU.

We loaded the trained models using TensorFlow and Scikit-learn. For CNN models, batch predictions were executed to leverage the GPU's parallel processing capabilities.

Tensorflow Model Implementation :

```
from tensorflow.keras.models import load_model
model = load_model('iris_model.h5')
predictions = model.predict(test_data)
```

SVM Exemple Model Implementation :

```
import joblib
import cupy as cp
svm_model_gpu = joblib.load('svm_model.pkl')

# Transfer data to GPU (using CuPy)
X_test_gpu = cp.asarray(X_test)

# Make predictions on GPU (convert input back to CPU because scikit-learn's S
y_pred_gpu = svm_model_gpu.predict(cp.asnumpy(X_test_gpu))
```

4.7.2 CPU Implementation

We are using the AMD Ryzen 5 4650G CPU : a high-performance APU (Accelerated Processing Unit), launched as part of AMD's Renoir series, which combines powerful Zen 2 architecture CPU cores with integrated Vega graphics. Originally targeted for OEM systems, this processor offers a blend of efficiency and capability suitable for both productivity tasks and light gaming.

It is built on 7nm technology, which allows for better power efficiency and performance compared to previous generations. The 4650G offers 6 cores and 12 threads, with a base clock of 3.7 GHz and a boost clock up to 4.2 GHz, making it versatile for multitasking and general-purpose workloads.

The CPU deployment involved loading models through the same Python environment but without GPU acceleration. All models (SVM, CNN, Random Forest) were tested sequentially due to the limited parallel capabilities compared to the GPU.

4.7.3 FPGA Implementation

For the FPGA Implementation, we used the FPGA PYNQ Z2, which is an open-source project from Xilinx that facilitates the use of these platforms. This board supports programming with Python and its various libraries, allowing developers to harness the advantages of programmable logic and microprocessors to create more innovative and promising electronic systems.

Instead of implementing IP cores or deploying the models as hardware accelerators, we used Python-based implementation directly on the PYNQ Z2 board via Jupyter notebooks. This method limits the acceleration potential of the FPGA, as the models are still executed sequentially on the FPGA's CPU without leveraging low-level hardware resources. The steps of the implementation are as follows :

- We begin by configuring an SD card by importing the PYNQ-Z image [96]. Next, we establish a connection between the PC and the board using an Ethernet connection and power the board with a USB cable.
- Then, we access the FPGA via a Jupyter Notebook using the IP address of the FPGA board [97].
- Import and then load the model, doing the predictions with python the same way we did with the CPU.

4.7.4 Results and Discussion

Model + Feature	Platform	Inference Time (s)	Power (W)	Memory (%)	Accuracy (%)
DNN + DWT (Symlet 8)	GPU (RTX 2060)	12	68	55	98.5
DNN + DWT (Symlet 8)	CPU (Ryzen 5)	72	30	40	98.5
SVM + DWT (Symlet 8)	GPU (RTX 2060)	4.2	35	45	97.2
SVM + DWT (Symlet 8)	CPU (Ryzen 5)	18	21	35	97.2
CNN (Direct)	GPU (RTX 2060)	20.5	58	50	97.0
CNN (Direct)	CPU (Ryzen 5)	111	38	50	97.0
KNN + DWT	FPGA (PYNQ Z2)	18	5.2	25	92.1
SVM + Gabor Filters	FPGA (PYNQ Z2)	25	4.5	20	94.3

Table 4.5: Performance Comparison Across Platforms: Inference Time, Power Consumption, Memory Usage, and Accuracy

The GPU outperformed the CPU in terms of inference time for all models due to its superior parallel processing capabilities. However, this came at the cost of higher energy consumption. The CNN with Symlet 8 DWT features performed best on the GPU, achieving high accuracy with low inference time.

The Ryzen 5 CPU offered a good trade-off between performance and energy consumption but couldn't match the GPU's speed. While classifiers like KNN and Random Forest were less impacted by the lack of parallelism, the CNN's performance suffered due to the heavier computational load. While not as fast, this platform is more energy-efficient and can be used in scenarios where power availability is limited.

The FPGA implementation showcased a great advantage in both time and power efficiency. The FPGA's hardware-level parallelism allows it to execute machine learning models more efficiently without the overhead of a general-purpose processor. However, since we used a higher-level implementation through Python notebooks, the FPGA's full potential was not fully utilized. The use of IP cores or running through HLS tools (High-Level Synthesis) to optimize latency, power, and memory usage could further reduce inference time and power consumption. using Python for implementation on the PYNQ Z2 resulted in a performance similar to that of the CPU, with no acceleration benefits from the FPGA's parallel architecture.

DWT-based methods (such as DWT + SVM) demonstrated their usefulness by offering a balance between accuracy and efficiency, proving particularly effective on CPUs with limited resources. While deep learning models like CNNs and DWT + DNN achieved slightly better accuracy, they did so at the cost of higher inference time, power, and memory consumption. The HVS-inspired DWT thus offers a practical advantage in real-world deployments where power

and computational resources are constrained, showcasing the benefits of biologically motivated feature extraction techniques over purely neural-network-based approaches.

4.8 Conclusion

In this chapter, we demonstrated the effectiveness of human visual system (HVS)-based methods for iris recognition, with the Discrete Wavelet Transform (DWT) delivering the highest accuracy, particularly with Symlet and Debauchies wavelets. Gabor filters also performed well but were limited by parameter sensitivity, while HOG and DCT fell short in capturing the complex iris textures. For classification, CNNs achieved the best results, leveraging their ability to learn hierarchical features, especially on GPU hardware, though at the cost of higher energy consumption. The FPGA implementation offered an optimal balance between performance and power efficiency, though its full potential could be unlocked with more advanced optimization techniques. Ultimately, our findings highlight that HVS-based feature extraction, especially using DWT, combined with efficient hardware implementations, is well-suited for high-precision iris recognition.

General Conclusion

This thesis has delved into the optimization of iris recognition using human visual system based methods, employing the HVS Based approachs to achieve superior accuracy and results. The comprehensive investigation undertaken in this study has provided significant insights into the Basics of the human Visual System physiology and the methods based on it leading to an efficient model.

In our work, we were specifically interested in iris recognition systems of other biometrics recognition system, because of their high performance in terms of precision. Such a system is mainly divided into two essential phases: detection and identification.

The study imployed many image processing techniques for the detection and preprocessing phase, beginning with the Canny Edge Detector aiming to, then the Hough transform which aims to locate the iris in an image and eliminate unnecessary information, keeping only the iris region in the form of a ring. Followed by The rubber sheet model of Daugman for the normalization step, in order to standardize the size of the images while retaining the essential discriminative information. This step transforms the iris image into a rectangular representation of uniform dimensions. We then focused on the crucial phase of extracting relevant features. For this, we applied Human Visual System Based models like the Gabor filters, which make it possible to exploit not only spatial information but also spectral information, which considerably improves the efficiency of recognition, as well as the Discrete Wavelete Transforms that provide both Time and Frequency Resolution and give a decomposition that works almost the same as the Human Visual System Modelisation : Decomposition of Perceptual Channels. We then applied different machine learning and deep learning based classifications like SVM, KNN and CNNs and compared the different accuracy.

Future research could explore the integration of additional HVS models, such as those that simulate contrast sensitivity or color perception, to enhance the feature extraction process for iris recognition. Future work could also focus on improving the robustness of iris recognition systems against occlusions and artifacts, such as reflections from eyeglasses or eyelids, by employing advanced image processing techniques. Also Developing explainable AI methodologies within the context of HVS-based iris recognition could facilitate greater transparency and trust in biometric systems, addressing potential concerns related to privacy and ethical considerations. And finally, Implementing the model on FPGA using the IP cores and HLS Modelisation to utilize the full potentiel of the FPGA.

In conclusion, this thesis focused on improving the iris recognition using Human Visual Based Systems combined with Deep learning and machine learning Classification methods.

Bibliography

[1] L Karthik Narayan, G Sonu, and SM Soukhya. Fingerprint recognition and its advanced features. *Int. J. Eng. Res. Technol*, 9(04):424–428, 2020.

[2] Shihab A Shawkat, Khalid Saeed Lateef Al-badri, and Ahmed Ibrahim Turki. The new hand geometry system and automatic identification. *Periodicals of Engineering and Natural Sciences*, 7(3):996–1008, 2019.

[3] Djamel Saigaa. *Contribution à l'authentification d'individus par reconnaissance de visages*. PhD thesis, Biskra, Université Mohamed khider. Faculté des Sciences et des Sciences de l'ingénieur, 2006.

[4] Yacine SAMAI. *Reconnaissance de l'Iris humain en utilisant les méthodes de l'Intelligence Artificielle*. PhD thesis, Université de Batna 2, 2012.

[5] Emine Krichen. *Reconnaissance des personnes par l'iris en mode dégradé*. PhD thesis, Evry, Institut national des télécommunications, 2007.

[6] Piotr Napieralski and Filip Rynkiewicz. Modeling human pupil dilation to decouple the pupillary light reflex. *Open Physics*, 17(1):458–467, 2019.

[7] Rebecca Engelberg, David Carrell, Elizabeth Krantz, Lawrence Corey, and Anna Wald. Natural history of genital herpes simplex virus type 1 infection. *Sexually transmitted diseases*, 30(2):174–177, 2003.

[8] Mohamed A El-Sayed and Mohammed A Abdel-Latif. Iris recognition approach for identity verification with dwt and multiclass svm. *PeerJ Computer Science*, 8:e919, 2022.

[9] Yusra Al-Najjar and Der Chen. Comparison of image quality assessment: Psnr, hvs, ssim, uiqi. *International Journal of Scientific and Engineering Research*, 3(8):1–5, 2012.

[10] Austin Roorda. Human visual system-image formation. *The encyclopedia of imaging science and technology*, 1:539–557, 2002.

[11] Gheyath Othman and Diyar Qader Zeebaree. The applications of discrete wavelet transform in image processing: A review. *Journal of Soft Computing and Data Mining*, 1(2):31–43, 2020.

[12] Urszula Marmol. Use of gabor filters for texture classification of airborne images and lidar data. *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, 22:325–336, 2011.

[13] Lynette I Millett and Joseph N Pato. Biometric recognition: Challenges and opportunities. 2010.

[14] Bogdan Kovalenko and Volodymyr Lukin. Usage of different chroma subsampling modes in image compression by bpg coder. *Ukrainian journal of remote sensing*, 9(3):11–16, 2022.

[15] Mehmet İsmail Gursoy, Seydi Vakkas Ustun, and Ahmet Serdar Yilmaz. An efficient dwt and ewt feature extraction methods for classification of real data pq disturbances. *International Journal of Engineering Research and Development*, 10(1):158–171, 2018.

[16] Scott J Daly. Visible differences predictor: an algorithm for the assessment of image fidelity. In *Human Vision, Visual Processing, and Digital Display III*, volume 1666, pages 2–15. SPIE, 1992.

[17] John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2(7):1160–1169, 1985.

[18] Yuping Feng, Shuang Chen, and Xuefeng Liu. Application of log-polar coordinate transform in image processing. In *2015 International Industrial Informatics and Computer Engineering Conference*, pages 1831–1835. Atlantis Press, 2015.

[19] Ljudmil Bojilov and Nadejda Bocheva. Neural network model for visual discrimination of complex motions. *Comptes rendus de l'Académie bulgare des Sciences*, 65(10):1379–1386, 2012.

[20] Guofa Li, Yifan Yang, Xingda Qu, Dongpu Cao, and Keqiang Li. A deep learning based image enhancement approach for autonomous driving at night. *Knowledge-Based Systems*, 213:106617, 2021.

[21] Ming Liu, Zhiqian Zhou, Penghui Shang, and Dong Xu. Fuzzified image enhancement for deep learning in iris recognition. *IEEE Transactions on Fuzzy Systems*, 28(1):92–99, 2019.

[22] Yuan-Tsung Chang, Timothy K Shih, Yung-Hui Li, and WGCW Kumara. Effectiveness evaluation of iris segmentation by using geodesic active contour (gac). *The Journal of Supercomputing*, 76:1628–1641, 2020.

[23] Jasem Rahman Malgheet, Noridayu Bt Manshor, and Lilly Suriani Affendey. Iris recognition development techniques: a comprehensive review. *Complexity*, 2021(1):6641247, 2021.

[24] Maryim Omran and Ebtesam N AlShemmary. An iris recognition system using deep convolutional neural network. In *Journal of Physics: Conference Series*, volume 1530, page 012159. IOP Publishing, 2020.

[25] Sundas Naqeeb Khan, Samra Urooj Khan, Onyeka Josephine Nwobodo, and Krzysztof Adam Cyran. Iris recognition through edge detection methods: Application in flight simulator user identification. *International Journal of Advanced Computer Science and Applications*, 14(4), 2023.

[26] Preeti Kamboj. Iris recognition segmentation signal processing system. *Applied Science & Engineering Journal for Advanced Research (2022) ISSN (Online)*, pages 2583–2468, 2022.

[27] Hind Hameed Rasheed, Sara Swathy Shamini, Moamin A Mahmoud, and Mohammad Ahmed Alomari. Review of iris segmentation and recognition using deep learning to improve biometric application. *Journal of Intelligent Systems*, 32(1):20230139, 2023.

[28] Sheng Lian, Zhiming Luo, Zhun Zhong, Xiang Lin, Songzhi Su, and Shaozi Li. Attention guided u-net for accurate iris segmentation. *Journal of Visual Communication and Image Representation*, 56:296–304, 2018.

[29] Wei Zhang, Xiaoqi Lu, Yu Gu, Yang Liu, Xianjing Meng, and Jing Li. A robust iris segmentation scheme based on improved u-net. *IEEE access*, 7:85082–85089, 2019.

[30] Alaa S Al-Waisy, Rami Qahwaji, Stanley Ipson, Shumoos Al-Fahdawi, and Tarek AM Nagem. A multi-biometric iris recognition system based on a deep learning approach. *Pattern Analysis and Applications*, 21:783–802, 2018.

[31] Hanaa M Ahmed and Mohammed A Taha. A brief survey on modern iris feature extraction methods. *Engineering and Technology Journal*, 39(1):123–129, 2021.

[32] Davis David. Random forest classifier tutorial: How to use tree-based algorithms for machine learning, Aug 2020.

[33] Kuo Wang and Ajay Kumar. Cross-spectral iris recognition using cnn and supervised discrete hashing. *Pattern Recognition*, 86:85–98, 2019.

[34] Andrej Hafner, Peter Peer, Žiga Emeršič, and Matej Vitek. Deep iris feature extraction. In *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 258–262. IEEE, 2021.

[35] Mohamed A El-Sayed and Mohammed A Abdel-Latif. Iris recognition approach for identity verification with dwt and multiclass svm. *PeerJ Computer Science*, 8:e919, 2022.

[36] Humayan Kabir Rana, Md Shafiul Azam, Mst Rashida Akhtar, Julian MW Quinn, and Mohammad Ali Moni. A fast iris recognition system through optimum feature extraction. *PeerJ Computer Science*, 5:e184, 2019.

[37] Pawel Drozdowski, Florian Struck, Christian Rathgeb, and Christoph Busch. Detection of glasses in near-infrared ocular images. In *2018 International Conference on Biometrics (ICB)*, pages 202–208. IEEE, 2018.

[38] Maria De Marsico, Alfredo Petrosino, and Stefano Ricciardi. Iris recognition through machine learning techniques: A survey. *Pattern Recognition Letters*, 82:106–115, 2016.

[39] Man Zhang, Zhaofeng He, Hui Zhang, Tieniu Tan, and Zhenan Sun. Toward practical remote iris recognition: a boosting based framework. *Neurocomputing*, 330:238–252, 2019.

[40] Muhammad Arsalan, Dong Seop Kim, Min Beom Lee, Muhammad Owais, and Kang Ryoung Park. Fred-net: Fully residual encoder–decoder network for accurate iris segmentation. *Expert Systems with Applications*, 122:217–241, 2019.

[41] Hugo Proença, Silvio Filipe, Ricardo Santos, Joao Oliveira, and Luis A Alexandre. The ubiris. v2: A database of visible wavelength iris images captured on-the-move and at-a-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1529–1535, 2009.

[42] Dongik Kim, Yujin Jung, Kar-Ann Toh, Byungjun Son, and Jaihie Kim. An empirical study on iris recognition in a mobile phone. *Expert systems with Applications*, 54:328–339, 2016.

[43] Hugo Proença and Luís A Alexandre. Ubiris: A noisy iris image database. In *Image Analysis and Processing–ICIAP 2005: 13th International Conference, Cagliari, Italy, September 6-8, 2005. Proceedings 13*, pages 970–977. Springer, 2005.

[44] M Dobes and Libor Machala. Upol iris database, 2004.

[45] Ajita Rattani and Reza Derakhshani. Ocular biometrics in the visible spectrum: A survey. *Image and Vision Computing*, 59:1–16, 2017.

[46] Tariq M Khan, Donald G Bailey, Mohammad AU Khan, and Yinan Kong. Real-time iris segmentation and its implementation on fpga. *Journal of Real-Time Image Processing*, 17:1089–1102, 2020.

[47] Kuo Wang and Ajay Kumar. Cross-spectral iris recognition using cnn and supervised discrete hashing. *Pattern Recognition*, 86:85–98, 2019.

[48] Yaosong Cheng, Yuanning Liu, Xiaodong Zhu, and Shuo Li. A multiclassification method for iris data based on the hadamard error correction output code and a convolutional network. *IEEE Access*, 7:145235–145245, 2019.

[49] S Rakshit. Novel methods for accurate human iris recognition. *University of Bath*, 2007.

[50] Saiyed Umer, Bibhas Chandra Dhara, and Bhabatosh Chanda. Iris recognition using multiscale morphologic features. *Pattern Recognition Letters*, 65:67–74, 2015.

[51] Ana F Sequeira, Joao C Monteiro, Ana Rebelo, and Hélder P Oliveira. Mobbio: A multimodal database captured with a portable handheld device. In *2014 International conference on computer vision theory and applications (VISAPP)*, volume 3, pages 133–139. IEEE, 2014.

[52] Muhammad Arsalan, Hyung Gil Hong, Rizwan Ali Naqvi, Min Beom Lee, Min Cheol Kim, Dong Seop Kim, Chan Sik Kim, and Kang Ryoung Park. Deep learning-based iris segmentation for iris recognition in visible light environment. *Symmetry*, 9(11):263, 2017.

[53] Yangyu Chen and Weigang Zhang. Iris liveness detection: A survey. In *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pages 1–7. IEEE, 2018.

[54] Vanaja Chirchi, Laxman Madhavrao Waghmare, and L Waghmare. Enhanced isocentric segmentor and wavelet rectangular coder to iris segmentation and recognition. *Int J Intell Eng Syst*, 10:1–10, 2017.

[55] Julian Fierrez, Javier Ortega-Garcia, Doroteo Torre Toledano, and Joaquin Gonzalez-Rodriguez. Biosec baseline corpus: A multimodal biometric database. *Pattern Recognition*, 40(4):1389–1392, 2007.

[56] Kuo Wang and Ajay Kumar. Cross-spectral iris recognition using cnn and supervised discrete hashing. *Pattern Recognition*, 86:85–98, 2019.

[57] Samuel P Fenker and Kevin W Bowyer. Analysis of template aging in iris biometrics. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 45–51. IEEE, 2012.

[58] Nianfeng Liu, Man Zhang, Haiqing Li, Zhenan Sun, and Tieniu Tan. Deepiris: Learning pairwise filter bank for heterogeneous iris verification. *Pattern Recognition Letters*, 82:154–161, 2016.

[59] Deepanshu Kumar, Mahati Sastry, and K Manikantan. Iris recognition using contrast enhancement and spectrum-based feature extraction. In *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pages 1–7. IEEE, 2016.

[60] Sushilkumar S Salve and SP Narote. Iris recognition using svm and ann. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 474–478. IEEE, 2016.

[61] Yixin Du, Thirimachos Bourlai, and Jeremy Dawson. Automated classification of mislabeled near-infrared left and right iris images using convolutional neural networks. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–6. IEEE, 2016.

[62] Yuniol Alvarez-Betancourt and Miguel Garcia-Silvente. A keypoints-based feature extraction method for iris recognition under variable image quality conditions. *Knowledge-Based Systems*, 92:169–182, 2016.

[63] Alaa S Al-Waisy, Rami Qahwaji, Stanley Ipson, Shumoos Al-Fahdawi, and Tarek AM Nagem. A multi-biometric iris recognition system based on a deep learning approach. *Pattern Analysis and Applications*, 21:783–802, 2018.

[64] Beeren Sahu, Pankaj Kumar Sa, Sambit Bakshi, and Arun Kumar Sangaiah. Reducing dense local feature key-points for faster iris recognition. *Computers & Electrical Engineering*, 70:939–949, 2018.

[65] Ramadan Gad, Muhammad Talha, Ahmed A Abd El-Latif, Mohamed Zorkany, EL-SAYED Ayman, EL-Fishawy Nawal, and Ghulam Muhammad. Iris recognition using multi-algorithmic approaches for cognitive internet of things (ciot) framework. *Future Generation Computer Systems*, 89:178–191, 2018.

[66] Soubhagya Sankar Barpanda, Pankaj K Sa, Oge Marques, Banshidhar Majhi, and Sambit Bakshi. Iris recognition with tunable filter bank based feature. *Multimedia Tools and Applications*, 77(6):7637–7674, 2018.

[67] Xiaonan Liu, Yuchen Bai, Yuwen Luo, Zhengwei Yang, and Yan Liu. Iris recognition in visible spectrum based on multi-layer analogous convolution and collaborative representation. *Pattern Recognition Letters*, 117:66–73, 2019.

[68] Kuo Wang and Ajay Kumar. Toward more accurate iris recognition using dilated residual features. *IEEE Transactions on Information Forensics and Security*, 14(12):3233–3245, 2019.

[69] Soubhagya Sankar Barpanda, Banshidhar Majhi, Panjak Kumar Sa, Arun Kumar Sangaiah, and Sambit Bakshi. Iris feature extraction through wavelet mel-frequency cepstrum coefficients. *Optics & Laser Technology*, 110:13–23, 2019.

[70] Kuo Wang and Ajay Kumar. Cross-spectral iris recognition using cnn and supervised discrete hashing. *Pattern Recognition*, 86:85–98, 2019.

[71] Narsi Reddy, Ajita Rattani, and Reza Derakhshani. Ocularnet: deep patch-based ocular biometric recognition. In *2018 IEEE international symposium on technologies for homeland security (HST)*, pages 1–6. IEEE, 2018.

[72] Zijing Zhao and Ajay Kumar. A deep learning based unified framework to detect, segment and recognize irises using spatially corresponding features. *Pattern Recognition*, 93:546–557, 2019.

[73] Young Won Lee, Ki Wan Kim, Toan Minh Hoang, Muhammad Arsalan, and Kang Ryoung Park. Deep residual cnn-based ocular recognition based on rough pupil detection in the images by nir camera sensor. *Sensors*, 19(4):842, 2019.

[74] Neda Ahmadi, Mehrbakhsh Nilashi, Sarminah Samad, Tarik A Rashid, and Hossein Ahmadi. An intelligent method for iris recognition using supervised machine learning techniques. *Optics & Laser Technology*, 120:105701, 2019.

[75] Min Beom Lee, Yu Hwan Kim, and Kang Ryoung Park. Conditional generative adversarial network-based data augmentation for enhancement of iris recognition accuracy. *IEEE Access*, 7:122134–122152, 2019.

[76] César Armando Estrebou, Martín Fleming, Marcos David Saavedra, and Federico Adra. A neural network framework for small microcontrollers. In *XXVII Congreso Argentino de Ciencias de la Computación (CACIC)(Modalidad virtual, 4 al 8 de octubre de 2021)*, 2021.

[77] Howard B.D Mark Hudson Beale, Martin T.H. Deep learning toolbox, 2020.

[78] Martin T.H Mark Hudson Beale. Deep learning hdl toolbox, 2020.

[79] Matlab for artificial intelligence, 1994-2023.

[80] Bing Liu, Yanzhen Zhou, Lei Feng, Hongshuo Fu, and Ping Fu. Hybrid cnn-svm inference accelerator on fpga using hls. volume 11. MDPI, 2022.

[81] Hossam O Ahmed, Maged Ghoneima, and Mohamed Dessouky. Concurrent mac unit design using vhdl for deep learning networks on fpga. In *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pages 31–36. IEEE, 2018.

[82] Hassan Kesserwani. The biophysics of visual edge detection: a review of basic principles. *Cureus*, 12(10), 2020.

[83] Anthony A Attama, Charles Lovelyna, and Ebele B Onuigbob. Nanotechnology for ocular and otic drug delivery and targeting. *Nanotechnology and drug delivery, Volume two: nano-engineering strategies and nanomedicines against severe diseases. CRC Press*, pages 165–90, 2016.

[84] Florent Autrusseau. *Modélisation psychovisuelle pour le tatouage des images*. PhD thesis, Université de Nantes, 2002.

[85] Kevin S Dingwell, Laurie C Doering, and David C Johnson. Glycoproteins e and i facilitate neuron-to-neuron spread of herpes simplex virus. *Journal of virology*, 69(11):7087–7098, 1995.

[86] Jeong-Sik Kim and Seung-Woo Lee. Peripheral dimming: A new low-power technology for oled display based on gaze tracking. *IEEE Access*, 8:209064–209073, 2020.

[87] Alexandre Benoît. *Le système visuel humain au secours de la vision par ordinateur*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2007.

[88] Jan Eichhorn. Applications of kernel machines to structured data. 2007.

[89] Richard E. Woods Rafael C. Gonzalez. Digital image processing. Pearson Education, Inc, 2008.

[90] Hakim Sénane. *Représentation d'images en sous-bandes visuelles: application au codage d'images de télévision sans défauts visibles*. PhD thesis, Nantes, 1996.

[91] Mayank Mishra. Convolutional neural networks, explained, Sep 2020.

[92] Svm classification.

[93] National Laboratory of Pattern Recognition.

[94] Rekrouk Maroua, Nbri Rihame, Allam Fatima Zohra, and Hamami Latifa. Study and implementation on fpga of human recognition system via iris based on deep learning. In *2023 International Conference on Decision Aid Sciences and Applications (DASA)*, pages 218–222. IEEE, 2023.

[95] Deepanshu Kumar, Mahati Sastry, and K Manikantan. Iris recognition using contrast enhancement and spectrum-based feature extraction. In *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pages 1–7. IEEE, 2016.

[96] Pynq-z2 image. www.pynq.io.

[97] Jupyter notebook pynq-z2. 192.168.2.99:9090.