




University of
Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: Computer Science/Reliable and Secure Systems	Spring semester, 20.23. Open / Restricted access
Writer: Mahboubeh Karimi	 (Writer's signature)
Faculty supervisor: Mina Farmanbar External supervisor(s):	
Thesis title: Frontal-Profile Face Recognition Using Deep Learning Algorithms	
Credits (ECTS): 30 ECTS	
Key words: Deep Learning, CNN, AlexNet, MLP, MTCNN, Haar Cascade, LBP.	Pages: 65 + enclosure: Appendix A Stavanger, 14.06.2023 Date/year



Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

Frontal-Profile Face Recognition Using Deep Learning Algorithms

Master's Thesis in Computer Science
by

Mahboubeh Karimi

Main Supervisor

Mina Farmanbar

Co-Supervisor

Muhammad Sulaiman

June 14, 2023

Abstract

This study investigates the impact of pose variation, particularly extreme poses such as the profile view, on the performance of biometric recognition systems. The study employs the Local Binary Pattern (LBP) approach in combination with convolutional neural networks (CNNs) to extract discriminative features from facial images. Feature-level and decision-level fusion techniques are utilized to enhance the system's performance. The experiments are conducted on the CFPW dataset, which consists of frontal and profile faces captured under diverse conditions. The results demonstrate that multimodal approaches outperform unimodal ones, with the fusion of frontal and profile images using the AlexNet model achieving the highest accuracy rate of 96.40%. This finding underscores the significance of incorporating multiple modalities, specifically frontal and profile images, to achieve robust and accurate face recognition. By combining these modalities, the system effectively mitigates the challenges posed by pose variation, resulting in improved recognition performance. The extraction of valuable features is crucial for the development of accurate face recognition systems. This study employs the LBP approach in conjunction with CNNs to extract discriminative features from facial images, enabling effective facial representation. To enhance the system's performance, feature-level and decision-level fusion techniques are employed. Feature-level fusion combines features acquired from both frontal and profile faces, while decision-level fusion combines classification decisions from individual classifiers. These fusion techniques leverage the complementary information provided by different modalities, improving overall recognition accuracy. The findings emphasize the effectiveness of multimodal approaches in biometric recognition systems. The utilization of multiple modalities, along with appropriate fusion techniques, enables the system to overcome limitations associated with pose variation and enhance the accuracy and reliability of face recognition. These insights contribute to the advancement of biometric recognition systems and open avenues for more robust and versatile applications in various domains.

Acknowledgements

I would like to express my sincere gratitude and appreciation to my supervisor, Professor Mina Farmanbar, for her invaluable guidance, support, and encouragement throughout the entire research process. Without her mentorship, this accomplishment would not have been possible. Her expertise, patience, and constructive feedback have been instrumental in shaping this thesis and elevating its quality. I wish to extend my thanks to Mr Muhammad Sulaiman, my co-supervisor for his continuous guidance, invaluable assistance, and support throughout the course of this study and research. I would like to express my gratitude to the technical and support personnel in the Electrical Engineering Department at the University of Stavanger for their valuable assistance and support throughout my master's studies. Lastly, I want to extend my heartfelt appreciation to my cherished family and friends, specifically my husband and beloved son, for their unwavering support, patience, and encouragement that played a crucial role in completing this project. Their presence and encouragement provided me with the strength and determination to overcome challenges and reach this milestone.

Contents

Abstract	iii
Acknowledgements	iv
1 Introduction	1
1.1 Problem background	2
1.2 Motivation behind the Research	3
1.3 Problem Statement and Research Questions	4
1.4 Outline	5
2 Related Work	9
3 Approach	15
3.1 Face Detection Methods	15
3.1.1 Haar Cascade Algorithm	15
3.1.2 Multi-Task Cascaded Convolutional Neural Networks (MTCNN)	18
3.2 Feature Extraction Methods	20
3.2.1 Local Binary Pattern (LBP)	20
3.2.2 Histogram of Oriented Gradients (HOG)	21
3.2.3 Principal Component Analysis Algorithm (PCA)	23
3.3 Face Recognition Techniques	26
3.3.1 Review of Convolutional Neural Network (CNN)	26
3.3.2 Multi-Layer Perceptron (MLP) classifier	28
3.3.3 AlexNet Classifier Architecture	30
4 Methodology	33
4.1 Dataset	34
4.2 pre-processing	35
4.3 Implementation	37
4.3.1 Data Pre-processing	38
4.3.2 Face Detection	38
4.3.3 Feature Extraction	40
4.3.4 Split Data into Train and Test Subsets	42
4.3.5 Dimensionality Reduction using PCA algorithm	43
4.3.5.1 PCA Projection	44

4.3.5.2	Finding Optimum Number of Principal Component . . .	44
4.3.6	Classification	46
5	Experimental Evaluation	51
5.1	Experimental Setup	51
5.2	Experimental Results	52
6	Conclusions	55
6.1	Future Directions	56
A	Poster	59
	Bibliography	61

Chapter 1

Introduction

In recent years, face recognition has emerged as the dominant biometric method in the fields of computer vision and pattern recognition. It finds widespread application in individual verification and identification in various practical domains, including finance, security and surveillance, commerce, and education. It is even integrated into consumer products such as mobile phones and social media platforms to provide user authentication and personalization. Despite numerous algorithms and procedures having been developed to improve the performance of face recognition, the complexities of this research area have made it a persistently challenging field [1].

The Overall Structure of the Face Recognition System. The face recognition systems generally consist of four main steps indicated in Figure 1.1. Face Detection, pre-processing, Feature Extraction, and Face Recognition [2]. This system uses an image as an input and the output is the identity of the individual identified in the given image.

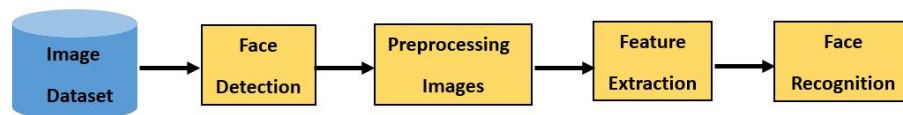


Figure 1.1: The basic architecture of the Face Recognition System

Face detection is identifying a specific area of an image as potentially containing a face. The facial features are acquired from the extracted face in the next stage. Finally, during the face classification process, the acquired features are compared with database values to identify a face. Many methods for face recognition are created by varying these four processes and combining them. Each of these processes will be briefly covered in the following paragraph:

- **Face Detection:** Facial detection is a critical component of facial recognition systems that seek to distinguish human faces from other objects within input images. This detection step is particularly useful in situations where input images contain multiple objects and individuals. Once a face is detected, image processing techniques can be applied to isolate and enhance the facial features, making subsequent facial recognition steps more accurate and effective.
- **Pre-processing:** During this phase, detected faces undergo processing to reduce noise and adjust for variations in illumination. Pre-processing, a crucial step in face recognition systems, involves a range of operations including image registration, scaling, face normalization, noise reduction, detection, and resizing. These procedures work in tandem to improve the accuracy of face recognition. Following pre-processing, the analysed faces can then undergo feature extraction to identify and extract distinctive facial features for use in subsequent recognition steps.
- **Feature Extraction:** The next phase in the face recognition process is feature extraction, which involves the application of powerful transformation techniques. This step involves reducing the dimensionality of the image while retaining significant features, resulting in a smaller yet still meaningful representation. The identification and isolation of crucial facial features that can be utilized for recognition or verification is a vital part of the feature extraction process. One approach to accomplish this is to use a feature extraction technique, which extracts and analyzes facial features at this stage. The resulting analysis transforms the image into a vector with a stable dimension and a set of constant points, representing the position of the extracted features.
- **Face Recognition:** The final step in the face recognition process is the classification phase, which utilizes robust classifiers such as fully connected neural networks and deep neural networks. This approach involves comparing the detected and processed face with a database of labelled faces to determine the identity of the person in question.

1.1 Problem background

Face verification has become a critical component in various domains, including security, surveillance, and mobile authentication. However, traditional face verification systems typically assume that faces are presented in frontal view, which is often not the case in real-world scenarios. In practice, faces are often presented at different angles, including profile views, which can make face verification challenging. This is particularly problematic

in unconstrained settings, where different pose conditions and facial expressions can further complicate face verification. Additionally, the challenge of face verification in the wild is exacerbated by the lack of standardization in image capture devices and lighting conditions. In outdoor settings, lighting conditions can be unpredictable, making it difficult to capture high-quality images.

The lack of accurate and reliable frontal-to-profile and in-the-wild face verification in such scenarios presents a significant challenge for the development of robust face recognition systems. This issue has significant implications for security and privacy, as inaccurate face verification can lead to false identification, leading to the wrongful conviction or misidentification of individuals. Therefore, there is a pressing need for research to address these challenges and develop more robust face verification systems that can accurately and reliably verify faces presented at different angles in real-world settings.

Moreover, there is a growing concern surrounding the potential for failure in face verification systems. Recent studies have revealed that facial recognition systems are prone to producing inaccurate or unreliable results, particularly when processing images with posture variations or distinct facial features. This issue can lead to misidentification, which can ultimately make the system unreliable. As a result, it is important to address this challenge by developing and implementing more robust and accurate face recognition technologies. Therefore, there is a pressing need for research to address these challenges and develop more robust face verification systems that can accurately and reliably verify faces presented at different angles in real-world settings. This requires the development of innovative techniques that can overcome the challenges posed by lighting conditions and environmental factors. It also involves addressing the issue of bias in face verification systems to ensure that they are fair and accurate for all individuals. By developing more accurate and reliable face verification systems, we can improve security and privacy in real-world scenarios.

1.2 Motivation behind the Research

Over the past decade, face recognition technology has evolved dramatically. Starting with limited, carefully obtained photos, the researchers have focused their attention on the different problems of face identification in unconstrained contexts. Face identification for unconstrained photos is a challenging task, due to variations in position, illumination, expression, age, and occlusion. When features of the entire face are not visible the challenge of the pose variation becomes more significant. These scenarios frequently occur in various real-world contexts, such as surveillance and photo tagging, when it is relatively normal for a person to avoid looking directly into the camera [3]. The majority

of face recognition systems have typically been created to identify faces from a frontal view, which is regarded as the most informative angle. However, because fewer facial features are visible in the profile view, it is more difficult to identify faces in this view [3].

Face detection and recognition systems rely on various methods, but they can be affected by factors such as pose, presence or absence of structural components, facial expression, occlusion, image orientation, imaging conditions, and time delay (for recognition). Many available applications developed by researchers are limited in their capabilities as they can typically only handle one or two of these effects, often with a narrow focus on specific well-structured applications. Developing a robust face recognition system that can work effectively under all conditions and encompass a wide scope of effects is challenging.

In summary, frontal-to-profile face recognition in uncontrolled environments is significant due to the following reasons:

- Frequent occurrence in various applications.
- Substantial degradation in performance of existing algorithms when comparing frontal faces to profile faces in real-world scenarios.
- Human performance in frontal-to-profile face comparisons is only marginally worse compared to frontal-to-frontal comparisons.

Overall, these factors emphasize the critical need for robust frontal-to-profile face recognition algorithms to address the challenges posed by uncontrolled environments, and further research in this area is warranted.

1.3 Problem Statement and Research Questions

Face verification poses a significant challenge due to the inherent variability in facial appearances caused by factors such as pose, illumination, expression, and occlusion. These variations can greatly impact the accuracy and reliability of face verification systems. Many existing face verification approaches primarily concentrate on comparing frontal face images, assuming ideal conditions and limited variations. However, such systems often struggle to perform effectively in real-world scenarios where face images are captured under diverse conditions and from different angles. To address this limitation, there is a growing demand for robust face verification systems that can handle face images captured in the wild, including those taken from various viewpoints and under challenging conditions. The development of such systems is crucial to ensure their practical applicability and reliability in real-world environments. By expanding the capabilities of face verification beyond frontal and profile face images, these systems

can enable accurate identification and verification of individuals captured in non-ideal conditions, such as surveillance footage, social media images, or images extracted from video streams.

This master thesis delves deeper into the following research questions, providing comprehensive answers and further insights:

1. How can we utilize machine learning and deep learning approaches to develop a face verification system that can verify face images captured from different angles in the wild?
2. What techniques can we use to address the issue of pose variation and improve the accuracy of face verification in the wild?
3. How can we effectively differentiate human faces from other objects within input images using face detection techniques?
4. How can we employ feature extraction methods that will yield more prominent features and accurate results?
5. How can we develop a robust face verification system that incorporates machine learning and deep learning techniques to handle illumination changes and poses variations in real-world scenarios?
6. How can we evaluate the performance of deep learning-based face verification systems in the wild, and what are their limitations?

This thesis aims to explore these questions by developing a face recognition model that leverages deep learning algorithms to accurately identify preprocessed human faces with variations in poses.

1.4 Outline

This section provides an overview of the thesis, outlining its structure and offering a concise summary of each chapter as listed below:

- **Chapter1: Introduction**

In this chapter, the thesis introduces its main objective, providing a comprehensive understanding of the topic and problem statement. It also offers a concise yet informative introduction to the field of face recognition technology and its diverse applications.

- **Chapter2: Related Work**

This chapter delves into an extensive exploration of accomplished studies and research that are directly relevant to face recognition methods. It critically examines previous works in the field, highlighting their methodologies, findings, and contributions to the advancement of face recognition technology. By analyzing the existing body of knowledge, this chapter sets the foundation for the subsequent chapters and establishes the context for the research conducted in this thesis.

- **Chapter3: Approach**

In this chapter, the thesis presents the main approaches and methods employed in the study. It introduces a comprehensive overview of the selected techniques, encompassing popular machine learning and deep learning algorithms that have been implemented. The chapter provides detailed explanations of these algorithms, highlighting their relevance to the research objectives and discussing their strengths and limitations.

- **Chapter4: Methodology**

In this chapter, the thesis focuses on presenting the dataset and methodologies employed in the study. The chapter provides a comprehensive overview of the dataset used, detailing its characteristics, size, and any preprocessing steps undertaken. Moreover, it highlights the various techniques and algorithms implemented to enhance the results of the study. The chapter discusses the rationale behind the selection of these methodologies and provides a clear description of their implementation process.

- **Chapter5: Experimental Evaluation**

In this chapter, the thesis thoroughly describes the evaluation of the methodologies employed to assess the effectiveness and performance of the proposed approach. The chapter provides a detailed explanation of the experimental setup, including the chosen evaluation metrics, the specific configurations of the algorithms, and any relevant parameters used. The chapter further presents the results obtained from the experiments conducted. This chapter contributes to the overall validity and reliability of the research, while also enabling the reader to gain a deeper understanding of the outcomes.

- **Chapter6: Conclusion**

This chapter provides a comprehensive summary of the work conducted throughout the research. It highlights the key findings, contributions, and implications of the study, emphasizing how they align with the initial objectives set forth in the introduction. The chapter also discusses potential future work that can address

any identified gaps or limitations in the research. It explores avenues for further exploration, suggesting areas where additional research can build upon the current findings. Furthermore, it explores promising approaches or methodologies that could be employed to overcome any challenges encountered during the study but were beyond the scope of the research.

Chapter 2

Related Work

Face recognition has been an active subject in the pattern recognition area. Recently, it has proceeded with the growth of CNNs. Moreover, CNNs have become the best solutions for various face recognition applications due to their outstanding abilities to represent and learn distinctive features. Before developing deep learning algorithms, most conventional face recognition techniques extracted shallow, hand-crafted features from facial photos, trained those features, and classified identities using Support Vector Machines (SVMs) or Nearest Neighbors (NNs) approaches. However, deep learning architectures have been developed and have produced incredibly excellent results for a variety of visual recognition tasks, including face recognition, because of the availability of cutting-edge computational capabilities and an increase in the availability of very large data sets. Massive research has made exceptional improvements in face recognition built on CNNs, and different methods have indicated remarkable enhancements in face recognition in the wild with pose variation. While several face recognition approaches have demonstrated promising outcomes under controlled conditions, the task remains exceptionally challenging in unconstrained environments. This difficulty arises due to the limited information available from single-face media investigations, especially when the quality of the images is low. Moreover, frontal-to-profile face verification has emerged as a highly active research area within computer vision and biometrics. In recent years, numerous studies and works have been dedicated to exploring this topic, aiming to overcome the challenges associated with verifying faces exhibiting frontal and profile views. The following section will delve into some of the most notable and relevant studies in this field.

In order to tackle the complex issue of face recognition under pose variation, researchers have explored and implemented various strategies. One commonly employed technique involves fitting a Morphable model to the face and subsequently warping it to a canonical

view. This approach has proven to be effective in mitigating the adverse effects of pose variation on face recognition accuracy.

Notably, the use of Generic Elastic Models (GEM) [4] and Active Appearance-based Models for Pose normalization [5] has gained significant traction in the field. These approaches, initially introduced in [6], have rapidly gained popularity as universal model fitting techniques for addressing pose variation in face recognition.

GEM provide a flexible framework for modeling and synthesizing facial variations caused by pose differences. By capturing and characterizing the shape and appearance changes associated with the pose, GEM allows for robust face normalization, enabling subsequent recognition algorithms to operate on a consistent and standardized face representation. Active Appearance-based Models (AAM) for Pose normalization offer another effective approach to handling pose variations. These models utilize a combination of shape and texture information to represent facial appearance under different poses. By aligning faces to a common reference shape and texture, AAM allows for pose-invariant comparisons and facilitates accurate recognition across varying pose conditions.

Both GEM and AAM techniques have proven successful in mitigating the challenges posed by face recognition with pose variation. They provide valuable tools for normalizing face images and establishing a canonical representation that is less susceptible to variations caused by different poses. As a result, these techniques enhance the accuracy and reliability of face recognition systems, contributing to the advancement of the field.

The continuous exploration and refinement of these techniques, along with the development of novel strategies, hold promising potential for further improvements in addressing pose variation and enhancing the overall performance of face recognition systems. While these techniques have shown promising results for faces with limited fluctuations and minor posture variations, they may not perform as well in real-world scenarios with more diverse facial expressions and poses. Another category of approaches that have been explored is subspace learning-based techniques. Canonical Correlation Analysis (CCA) [7] and Partial Least Square (PLS) [8] are examples of commonly used subspace learning techniques in this domain. These techniques have shown the potential in improving the robustness of face recognition systems to variations in poses and expressions encountered in real-world situations.

Two recent studies, [9] and [10], have demonstrated effective results in recognizing faces with varying poses using identification-based approaches on datasets such as MultiPie and CMU PIE. In particular, [9] achieved recognition accuracy of 27.1% for Frontal Profile in MultiPie. However, it is important to note that the effectiveness of these techniques in recognizing faces in wild situations, where images are not controlled, has not yet been

proved. Further research is required to evaluate their performance in such scenarios. Generative models have been a focus of research in the field of pose-invariant face recognition. These techniques assume that a latent variable is responsible for generating various identities and poses via a latent factor.

In recent years, [11] has shown solid performance in constrained data sets such as FERET [12]. Similarly, [13] has demonstrated impressive outcomes on unconstrained datasets such as LFW, achieving a validation accuracy of 90.07% in an unrestricted scenario. These results are highly encouraging and point towards the potential of generative models in face recognition. Another approach to addressing the issue of pose variation in face recognition is attribute-based recognition [14]. This technique has the potential to be invariant to posture changes, but it remains unclear whether features can be accurately determined on profile faces as they are on frontal faces. However, in the case of our suggested CFP dataset, achieving strong alignment across poses, which is necessary for many of these approaches, is challenging.

In addition to the pose variation problem, this section also covers approaches that have demonstrated success in unconstrained settings and ‘in the wild’ data sets like LFW. In order to address the variability of unconstrained images, several researchers have developed metric learning algorithms that can learn a transformation of the feature space. In particular, the LFW dataset has yielded promising results for Cosine Similarity metric learning [15] and Similarity metric learning [16], achieving an accuracy of 86.73% under unrestricted circumstances. These results highlight the potential of metric learning algorithms for face recognition in real-world scenarios.

Along with Deep metric learning methodologies [17] [18], other metric learning approaches have also been developed by researchers [19], [20]. Performance on LFW is high for the Joint Bayesian models [21] (90.90% accuracy in an unrestricted scenario) and [22] (93.18% accuracy). These techniques can only be utilized with the unconstrained protocol (where identity information or outside training data can be used) because they typically require identification information during training.

To provide a more comprehensive overview, researchers have explored feature extraction methods beyond the conventional SIFT, LBP, or HoG. Fisher Vector encoding [23] (achieving 87.47% accuracy in constrained conditions) and [24] (achieving 84.08% accuracy in restricted settings) are both effective approaches, but they lack robustness to significant pose variation. In recent years, researchers have shifted from hand-crafted features to trained features by leveraging CNNs and deep networks. Notable examples of successful applications include Deepface [25] (97.35% accuracy), DeepID [26] (99.47% accuracy), and FaceNet [27] (99.63% accuracy), which represent the state-of-the-art in face recognition on LFW in the unrestricted context with outside training data.

In their study, the authors in [3] have assembled Celebrities in Frontal-Profile (CFP) face data collection specifically designed to facilitate research on frontal-to-profile face verification in real-world scenarios. This dataset aims to isolate and explore the factor of pose variation, which is particularly challenging for extreme positions like profiles where many facial features are obscured. Additionally, the dataset encompasses other variations commonly encountered in unconstrained environments, thereby simulating "in the wild" conditions.

During their experimental evaluation, the authors made an intriguing discovery. They found that when humans performed frontal-profile verification, their accuracy was only slightly lower at 94.57% compared to frontal-frontal verification, which achieved an accuracy of 96.24%. This finding implies that humans exhibit a relatively robust ability to verify faces across varying poses, even under challenging profile conditions.

However, when several state-of-the-art algorithms were subjected to the same evaluation, including Deep learning algorithms, Fisher Vector, and Sub-SML, a significant performance drop was observed from frontal-frontal to frontal-profile verification. In fact, each algorithm experienced a decrease in accuracy of more than 10%. Notably, the Deep learning implementation exhibited a substantially lower accuracy of 84.91% on frontal-profile verification, in contrast to frontal-frontal verification, where it performed comparably to human performance.

These results highlight a notable performance gap between automatic face recognition techniques and human performance when confronted with substantial posture variations in unrestricted photographs. The disparity in accuracy indicates the inherent difficulty faced by automatic algorithms in effectively handling pose variations and recognizing faces under challenging conditions.

The findings from this study shed light on the limitations of current state-of-the-art algorithms and emphasize the need for further research and innovation in developing more robust face recognition techniques that can bridge the performance gap with human abilities. By addressing the challenges posed by pose variations in real-world scenarios, future advancements in automatic face recognition systems can aim to achieve performance levels comparable to or even surpassing human performance, thus enhancing their applicability and effectiveness in a wide range of practical applications.

In their experiments, the authors found that Fisher Vector performs better than HoG and LBP among various hand-crafted features. Particularly, when combined with metric learning SubSML, Fisher Vector achieves remarkable accuracy of 80.63% on Frontal-Profile and 91.3% on Frontal-Frontal datasets, even in restricted settings. These results highlight the continued difficulty of face recognition in uncontrolled environments with

significant pose variations, emphasizing the importance of continued research in this field [3].

Chapter 3

Approach

In this section, the main approaches and methods used in this study are introduced, including the popular machine learning and deep learning algorithms that have been implemented. As outlined in the introduction, the face identification framework has three main components: face detection, feature extraction, and face recognition. In the following sections, a brief overview of each of these components will be provided before delving into the details.

3.1 Face Detection Methods

Face detection is a crucial component of the face identification process and serves as the first step in face recognition. It falls under the domain of computer vision technology, which involves detecting the position and dimensions of facial images within a digital photograph while ignoring other objects in the image. This section will briefly overview the techniques used to detect human faces and facial landmarks.

3.1.1 Haar Cascade Algorithm

The Haar cascade classifier, initially proposed by Viola and Jones in their seminal 2001 publication, Rapid Object Detection using a Boosted Cascade of Simple Features [28], is OpenCV's most common object detection algorithm. It is an efficient pre-trained machine-learning algorithm used to detect faces in an image or a real-time video because it is trained from a great quantity of positive and negative pictures. This paper uses Haar cascade detection with an open Computer Vision library (open cv) to identify human faces. The Haar cascade method eliminates most of the irrelevant features from images and provides a unique human face. Thus, it will enhance the accuracy of the

face recognition model. The Figure 3.1 represents the flow system of the Haar cascade classifier.

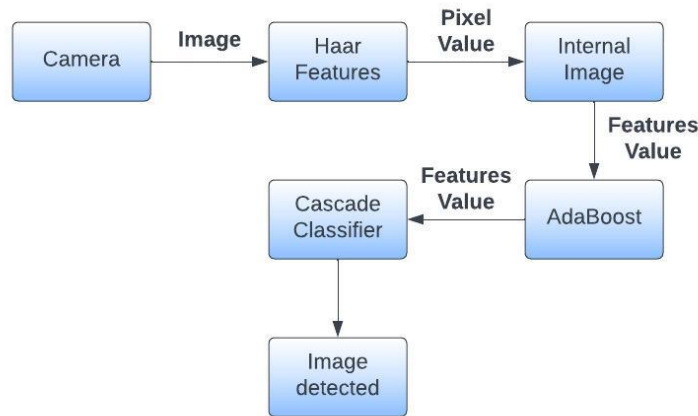


Figure 3.1: The flow system of the Haar cascade classifier

The Haar cascade algorithm for face detection can be described in three main phases:

1. Haar Feature Selection

The first step is to select the Haar-like features that most effectively detect faces. These features are rectangular regions of an image that are used to measure the intensity differences between adjacent areas of the image. Some examples of Haar-like features include edge features, line features, and four-rectangle features are shown in Figure 3.2 [29].

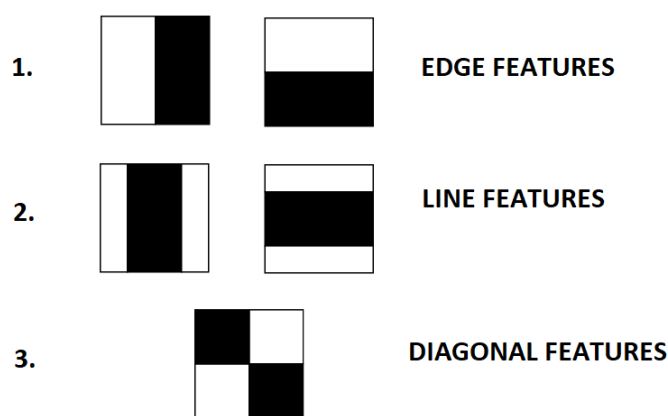


Figure 3.2: Some examples of Haar features [29].

The algorithm uses a machine-learning technique called AdaBoost to select the most significant features. AdaBoost trains multiple weak classifiers on the training data, with

each classifier using a different set of Haar-like features. The weak classifiers are then combined to form a strong classifier, which is used to detect faces in new images.

2. Integral Image Calculation

The next step is to calculate the integral image of the input image. The integral image is a 2D array that stores the sum of all the pixels in the input image up to a given pixel. This calculation can be done efficiently using dynamic programming. The integral image is used to speed up the calculation of Haar-like features. Instead of computing the sum of pixel intensities for each feature in the image, the algorithm uses the integral image to calculate the sum of pixel intensities for each feature in constant time. The integral image and Haar-like rectangle features are illustrated in Figure 3.3 [30].

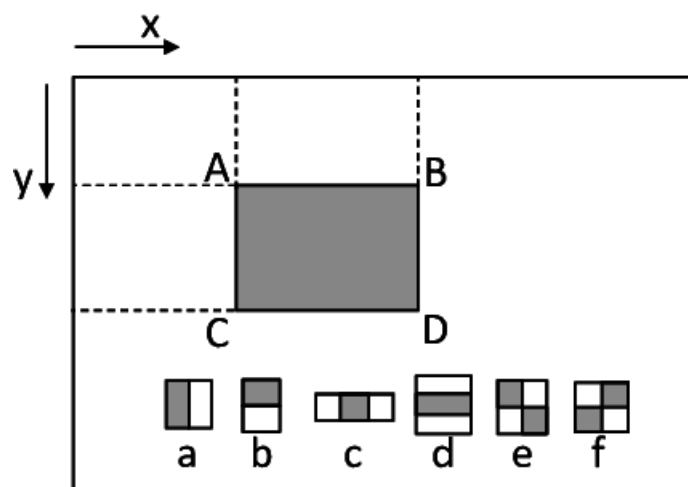


Figure 3.3: Illustration of the integral image and Haar-like rectangle features [30].

3. Implementing Cascading Classifiers

The final step is to use the Haar-like features and the integral image to detect faces in the input image. The algorithm uses a cascade classifier, a sequence of classifiers trained to progressively eliminate non-face regions of the image. Each classifier in the cascade is trained to be highly selective and has a low false positive rate. The cascade classifier operates in stages, with each stage using a different set of Haar-like features to identify regions of the image that may contain a face. At each stage, the classifier calculates a feature vector for each region of the image using the integral image. If the feature vector matches the feature vector of a face, the region is passed to the next stage of the cascade. Otherwise, the region is rejected as non-face. The cascade classifier can be trained on many positive and negative samples to improve its accuracy. The classifier can also be fine-tuned to detect specific types of faces, such as faces in different lighting conditions, orientations, and expressions [30].

3.1.2 Multi-Task Cascaded Convolutional Neural Networks (MTCNN)

Facial landmark detection is a crucial step in many facial analysis tasks, including facial recognition, expression analysis, and virtual try-on systems. This process involves identifying and locating specific points on a face, such as the corners of the eyes, nose, and mouth. One popular face detection algorithm that can also perform facial landmark detection is MTCNN. Introduced in 2016 by Zhang et al [31], MTCNN is a deep learning architecture that comprises three neural networks that work together to identify both faces and facial landmarks in images. MTCNN has been proven to be an efficient and accurate approach for detecting facial landmarks in various applications. MTCNN is comprised of three networks, namely the Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net). Each network has a specific task in the face detection process, as outlined below:

- **Proposal Network (P-Net):** The P-Net is the first network in the cascade and is responsible for generating candidate face regions (face proposals) from an input image. It uses a fully convolutional neural network to scan the entire image and output a set of bounding boxes that potentially contain faces. The network takes the entire input image as input and produces a set of candidate boxes with different sizes and aspect ratios.
- **Refine Network (R-Net):** The Refine Network (R-Net) is the second network in the MTCNN cascade. Its main objective is to enhance the accuracy of face detection by filtering out false positives and refining the bounding boxes produced by the Proposal Network (P-Net). R-Net takes the candidate boxes generated by P-Net as input and produces refined bounding boxes that are closer to the actual face regions. Like P-Net, R-Net is also a fully convolutional neural network. However, unlike P-Net, it takes candidate boxes rather than the entire image as input. The use of bounding boxes helps in cropping out unnecessary parts of the image, eliminating the background and focusing only on the distinctive face region of the image.
- **Output Network (O-Net):** The O-Net is the final network in the cascade and its primary task is to further refine the bounding boxes produced by the R-Net and to classify them as face or non-face regions. This network takes the refined bounding boxes produced by the R-Net as input and outputs the final set of bounding boxes and facial landmarks. The O-Net is also a fully convolutional neural network, but it is more complex than the previous two networks and can detect finer facial features such as the eyes, nose, and mouth.

Figure 3.4 describes the architecture of the layers applied in each step of the cascaded MTCNN model [32].

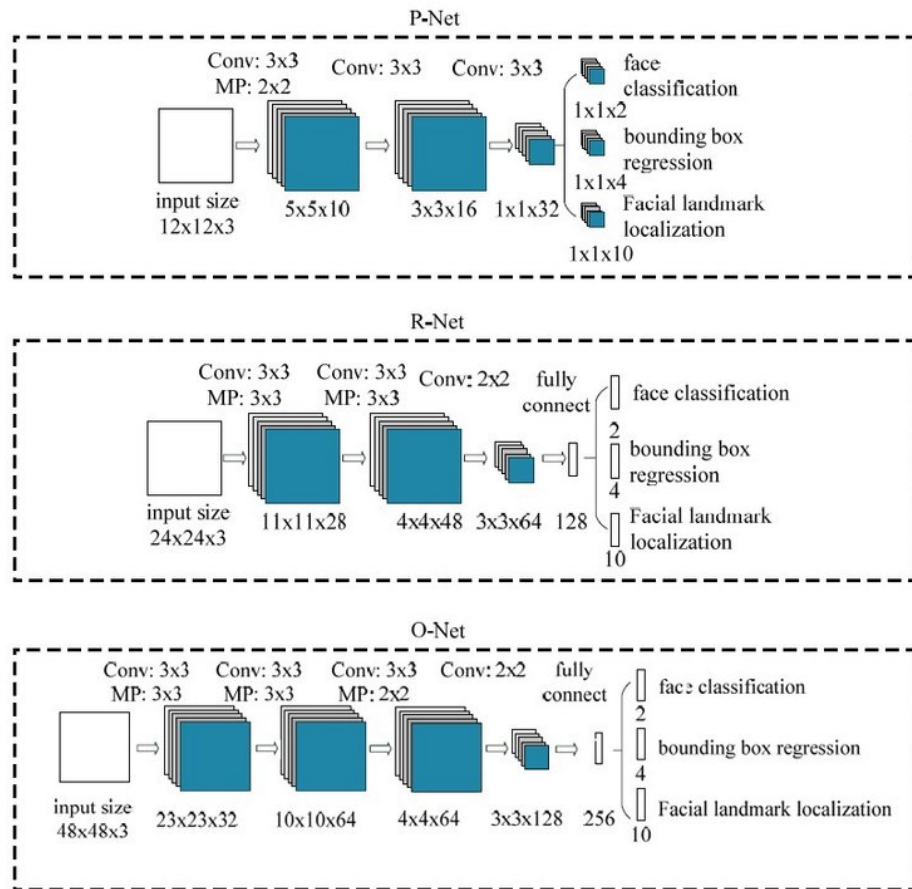


Figure 3.4: Stage architecture of the MTCNN model used for face detection and landmark extraction [32].

As already explained, MTCNN preserves three tasks including facial landmark localization, bounding box regression, and classification of faces. To create the same class of results, each stage employs a different number of layers and a different size of Conv. filters. Three categories describe the outputs. The first set of outputs uses two neurons: one to detect the presence of a face and the other to calculate its classification score. Bounding box regression, which represents the top left and lower right of the bounding box by four neurons, is another component of the output. Five sites on the left eye, right eye, nose, left mouth corner, and right mouth corner are regressed by facial landmark localization. This means that ten neurons are required to represent the 10-D variable [32].

3.2 Feature Extraction Methods

Features in images are defined as significant local intensity changes when shifted over an image. Edges and corners are notable examples of image features. Feature descriptor methods, including edge detection, have been employed in various applications, such as object detection, face recognition, image segmentation, and region separation. In the context of face recognition, facial features such as eyes and nose are considered significant. Feature extraction methods aid the classifier model in distinguishing between different individuals' faces by extracting effective and prominent information from images. The face recognition system categorizes images based on the value of simple features. There are several reasons for utilizing features instead of pixels directly. The most prominent reason is that features can encapsulate ad-hoc domain information, which is challenging to learn using a finite amount of training data. The second crucial incentive for using features is that the feature-based system operates significantly faster than a pixel-based system [28]. In the following section, some feature extraction methods that are applied to the dataset images are briefly introduced.

3.2.1 Local Binary Pattern (LBP)

LBP is a method utilized in image processing and computer vision for extracting features. In 1994, Ojala et al. presented the method, which has subsequently become extensively employed in various applications, including facial recognition, texture classification, and object detection, owing to its simplicity and robustness. The LBP algorithm is used to represent the texture features of images at a local level and has the advantage of being insensitive to rotation and grayscale levels. It is a crucial method for identifying features in an image and can cope with lighting variations.

The LBP method is a simple yet effective way to describe the local texture of an image. It involves comparing each pixel in an image with its neighboring pixels and encoding the result as a binary pattern. Specifically, a binary code is generated for each pixel based on whether the surrounding pixels have a higher or lower intensity value than the center pixel. This binary code is then used as a feature for that pixel [33]. In simple terms, the LBP technique segments a facial image into multiple regions, extracts LBP features from each region, and combines them to create a feature vector that represents the facial descriptor [34]. To assign a binary label to each pixel in an image, the LBP operator uses a 3×3 neighborhood surrounding that pixel and thresholds it with the center pixel value. The resulting binary code is then interpreted as a binary integer.

In other terms, LBP is described as an ordered sequence of binary comparisons of the pixel intensities of the core pixel and its surrounding pixels for a given pixel position (x, y) . The 8-bit word's final decimal label value can be stated in the equation 3.1 as follows [34] and [35]:

$$LBP(x, y) = \sum_{n=0}^7 2^n \cdot S(L_n(x, y) - L_c(x, y)) \quad (3.1)$$

where the grey value of the centre pixel (x, y) is represented by L_c , the grey values of the 8 surrounding pixels are denoted by L_n , and function $S(k)$ is expressed as:

$$S(k) = \begin{cases} 1 & \text{if } k \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

The value of this centre pixel serves as a threshold for the LBP operator, which operates on a pixel's eight neighbours [36]. A neighbouring pixel receives one if its grey value is higher (or equal) to the centre pixel, otherwise, it receives zero. The eight ones or zeros are then combined to form a binary code, which creates the LBP code for the centre pixel as illustrated in Figure 3.5.

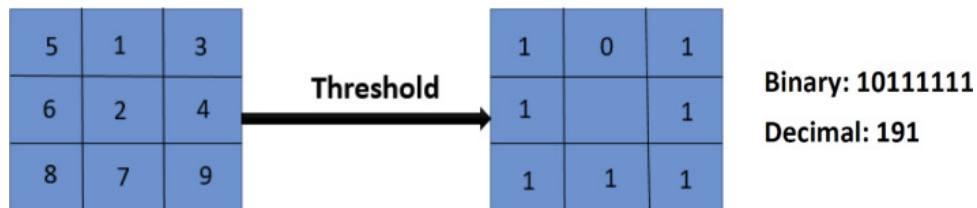


Figure 3.5: The basic LBP operator which labels the pixels in the image [33].

The present study employs LBP as a method for converting acquired images into binary vectors. The resulting facial feature vectors retain the image pixels within a predetermined threshold. These vectors are then combined with weights to create a network architecture pattern for facial classification, utilizing CNN models.

3.2.2 Histogram of Oriented Gradients (HOG)

HOG is a popular feature extraction method used in computer vision and image processing tasks, particularly in object detection and recognition. The main idea behind the HOG feature extraction method is to capture and describe the local shape and structure of an image by analyzing the distribution of gradient orientations. Gradients represent the changes in intensity values across an image, and they provide important information

about the edges and boundaries of objects. A step-by-step explanation of the HOG feature extraction method and its structure is mentioned below:

1. Gradient Computation:

- Calculate the gradients (derivatives) of the image in both the horizontal and vertical directions.
- Typically, the Sobel operator is used to compute the gradients, resulting in two gradient images: one for the horizontal gradients (G_x) and one for the vertical gradients (G_y).
- From the horizontal and vertical gradients, the magnitude and orientation of the gradient vectors are calculated.

2. Gradient Orientation Binning:

- Divide the image into small cells (e.g., 8x8 pixels).
- For each cell, accumulate the gradient orientations into a histogram.
- The histogram has a predefined number of bins, usually representing a range of angles (e.g., 0-180 degrees).
- The magnitude of each gradient is also considered and contributes to the corresponding bin.

3. Histogram Normalization:

- Normalize the histograms within a larger block of cells (e.g., 2x2 or 3x3 cells).
- This normalization is performed to enhance the robustness of the features to changes in illumination and contrast.
- Common normalization methods include L1-norm or L2-norm normalization.

4. Feature Descriptor:

- Concatenate all the normalized histograms from the previous step to form a feature vector.
- The resulting feature vector represents the local structure and shape information of the image.

- The length of the feature vector depends on the number of cells, bins per histogram, and the size of the blocks.

5. Sliding Window:

- Apply a sliding window across the entire image to extract HOG features at different spatial locations.
- The sliding window moves in predefined strides and scales to capture features at various scales.

The HOG feature extraction method effectively captures the local shape information and is particularly robust against changes in illumination and contrast. It has been widely used in various computer vision applications, including pedestrian detection, face detection, and object recognition in images. The structure of the HOG descriptor is shown in figure 3.6 [37]:

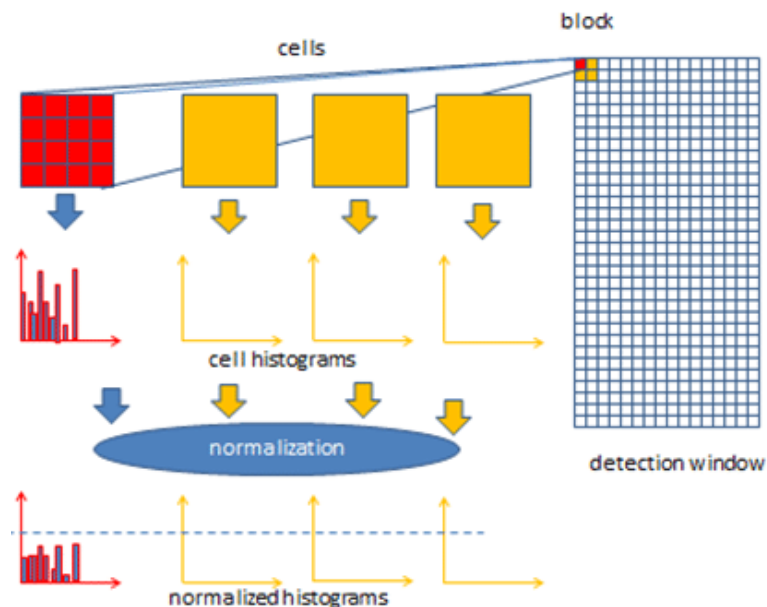


Figure 3.6: HOG Implementation [37].

3.2.3 Principal Component Analysis Algorithm (PCA)

PCA is a statistical technique that is widely employed for reducing the dimensionality of data, extracting features, and compressing information. It involves a linear transformation approach that enables the identification of patterns in high-dimensional data by projecting

it onto a lower-dimensional space while preserving most of the original variance. In face recognition applications, it can be utilized to extract significant features from images that aid in person identification. The basic principle underlying PCA for face recognition is to identify the most essential facial features that accurately represent the face images in a lower dimensional space [38]. There are several reasons why PCA is suitable for face recognition:

1. **Dimensionality reduction:** Face images often contain a high number of pixels, resulting in a large feature space. PCA helps reduce the dimensionality by transforming the original data into a lower-dimensional representation, known as eigenfaces. This reduces computational complexity and memory requirements while preserving essential facial features.
2. **Face representation:** PCA captures the most significant variations in face images by identifying a set of eigenfaces, which are orthogonal vectors representing the principal components of the face data. These eigenfaces are computed by analyzing the covariance matrix of the face image dataset. They provide a compact representation of faces, allowing for efficient face recognition.
3. **Discriminative power:** PCA focuses on capturing the maximum variance in the data. In the context of face recognition, this means that the eigenfaces obtained from PCA tend to represent the most discriminative facial features. By projecting a new face image onto the eigenface subspace, the algorithm can effectively identify and match faces based on these discriminative features.
4. **Robustness to variations:** It is relatively robust to variations in lighting conditions, pose, and facial expressions, as it captures the underlying structure of face images rather than relying on specific pixel intensities. This enables effective face recognition even in the presence of moderate variations in the input images.

In summary, PCA provides a powerful dimensionality reduction technique for face recognition applications by extracting discriminative features and enabling efficient matching and identification.

This section presents an overview of the theory of the PCA algorithm in face recognition and outlines the flow chart for the PCA algorithm, as depicted in Figure 3.7 [39].

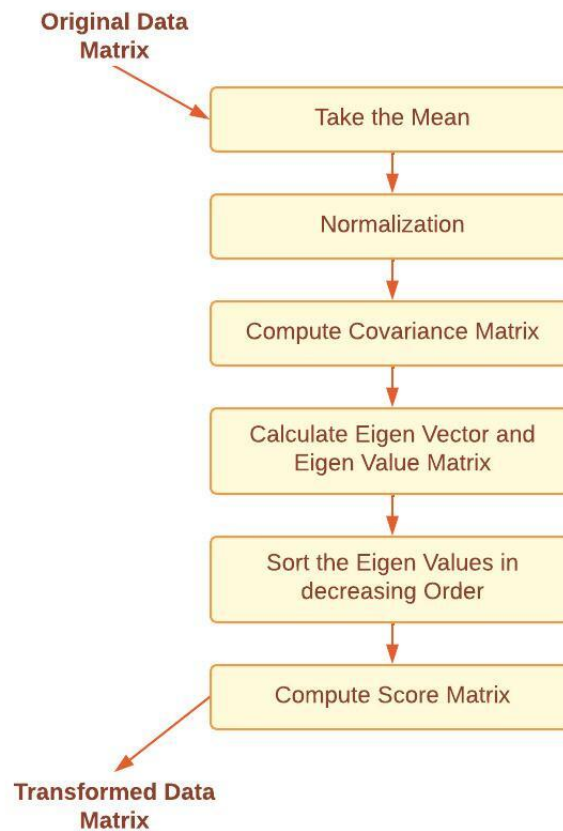


Figure 3.7: The flow chart for the PCA algorithm.

As it is displayed in the flow chart the steps involved in the PCA algorithm are as follows:

- **Normalization:** This is the first step in PCA, which entails normalizing the data by subtracting the mean and dividing it by the standard deviation. This preliminary stage is particularly critical, as PCA is known to be affected by the scaling of variables.
- **Covariance Matrix:** computing the covariance matrix of the normalized data is the next stage. The covariance matrix is a square matrix that includes the covariances between all pairs of variables. It is employed to determine the linear correlation between variables.
- **Eigenvectors and Eigenvalues:** The next step is to calculate the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors represent the directions of maximum variance in the data, while the eigenvalues quantify the degree of variance accounted for by each eigenvector. The eigenvectors are also called the principal components.

- **Selecting the Principal Components:** The next step is to choose the principal components that describe the most variance in the data. This can be done by ranking the eigenvalues in descending order and selecting the top k eigenvectors that explain the most variance, where k is the desired dimensionality of the new feature space.
- **Projection:** The final step is to project the data onto the new feature space defined by the selected principal components. This is done by multiplying the normalized data by the eigenvectors corresponding to the selected principal components. The resulting matrix contains the transformed data in the lower-dimensional space.

3.3 Face Recognition Techniques

The recognition phase of a face identification system is the final stage where the system analyzes and matches the detected face with a known database of labelled faces. In this process, the system then compares extracted features with the corresponding features of faces in its database to determine the identity of the person in question. This process involves using algorithms and classifiers such as fully connected neural networks and deep neural networks, which have been trained to identify facial features and patterns. Overall, the recognition phase is crucial in ensuring that the face identification system can accurately and reliably determine the identity of a person based on their facial features. This section will provide, a brief overview of the techniques used in the face recognition system to determine the identity of the individual in question.

3.3.1 Review of Convolutional Neural Network (CNN)

CNNs are a type of neural network that excels at tasks related to classifying and identifying images. They are multi-layered feed-forward neural networks that consist of filters, kernels, or neurons with biases, parameters, and trainable weights. Each filter processes a set of inputs through convolution and, optionally, nonlinearity. Overfitting can occur in a CNN model when the neural network is complex, and the input data is small. A high prediction accuracy is achieved if the model has a low loss function on the training data. However, overfitting occurs when the loss function is large on the test data, resulting in low prediction accuracy. On the other hand, underfitting may occur if the model performs poorly on the training data, as the model cannot capture the relationship between the input examples and the target values. CNNs can be trained using supervised learning, where the network is given a set of labeled images and learns to recognize the patterns and features in the input images. They can also be trained

using unsupervised learning, where the network learns to extract useful features from the input images without any labeled data. Overall, CNNs have proven to be a powerful tool in the field of image processing and computer vision, allowing for accurate and efficient image classification and identification.

A typical CNN architecture is depicted in Figure 3.8 [40]. The structure of CNN consists of Convolutional, Pooling, Rectified Linear Unit (ReLU), and Fully Connected Layers.

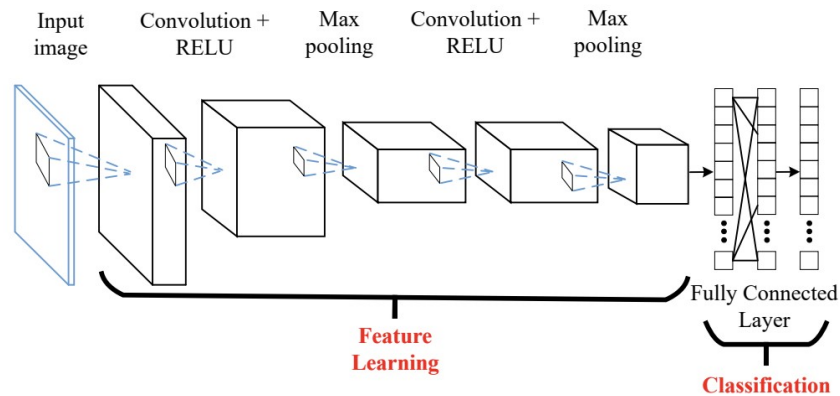


Figure 3.8: The basic architecture of the CNN model [40].

- **Convolutional Layer:** The convolutional network's fundamental component, which handles most of the computational tasks, is the convolutional layer. The main objective of the convolution layer is to extract features from the input data, which is a picture. Convolution learns visual attributes from tiny squares of input images, maintaining the spatial correlation between pixels. By using a group of learnable neurons, the input image is distorted. As a result, the output image contains a feature map or activation map, which is then used as input data for the subsequent convolutional layer [40].
- **Pooling Layers:** The pooling layer is a key component in convolutional neural networks (CNNs) that reduces the spatial size of the input image through down-sampling. Typically, it is applied after a convolutional layer to reduce the spatial resolution of the feature maps and enhance the network's efficiency. The most widely used type of pooling is max pooling, which selects the maximum value in each local region of the feature map. By adding this layer between convolutional layers, the network can achieve higher generalization, faster convergence, and greater robustness to translation and distortion.
- **ReLU Layer:** ReLU is a non-linear operation that comprises rectifier-using units. Since it is an element-wise procedure, each pixel is affected, and all negative values

in the feature map are replaced with zero. To comprehend how the ReLU works, we assume that x is the neuron input, and the rectifier is defined as $f(x) = \max(0, x)$ in the literature for neural networks.

- **Fully Connected Layer:** The fully Connected Layer (FCL) describes every filter in the previous layer as being connected to every filter in the following layer. High-level features of the input image are represented in the output from the convolutional, pooling, and ReLU layers. Utilizing these features to divide the input image into different classes depending on the training dataset is the aim of using the FCL. It is considered the last pooling layer that delivers the features to a classifier that utilizes the SoftMax activation function. The sum of output possibilities from the Fully Connected Layer is 1. Using SoftMax as the activation function guarantees this. The SoftMax function reduces a vector of arbitrary real-valued scores to a vector of values that vary from zero to one and add to one [41]. A SoftMax function is applied to the output of the final fully connected layer to produce a probability distribution over the output classes. By minimizing the cross-entropy loss between the predicted probability and the true labels during training, the network is optimized.

3.3.2 Multi-Layer Perceptron (MLP) classifier

MLP is a popular type of artificial neural network used in various applications of machine learning and pattern recognition. Also known as feedforward neural networks, MLPs propagate the input signal through a sequence of layers without loops or cycles between them. They comprise an input layer, one or more hidden layers, and an output layer. The input layer receives input data, and the output layer produces the output. The hidden layers are where most of the computation takes place. Each neuron in a layer is connected to every neuron in the next layer, with weights assigned to each connection. The number of neurons in each layer can be adjusted to fit the problem at hand. Each neuron uses an activation function, such as sigmoid, ReLU, or tanh, to transform its input into an output. This activation function adds non-linearity to the MLP, enabling it to learn complex functions. Figure 3.9 [42] depicts an example of an MLP model with one hidden layer.

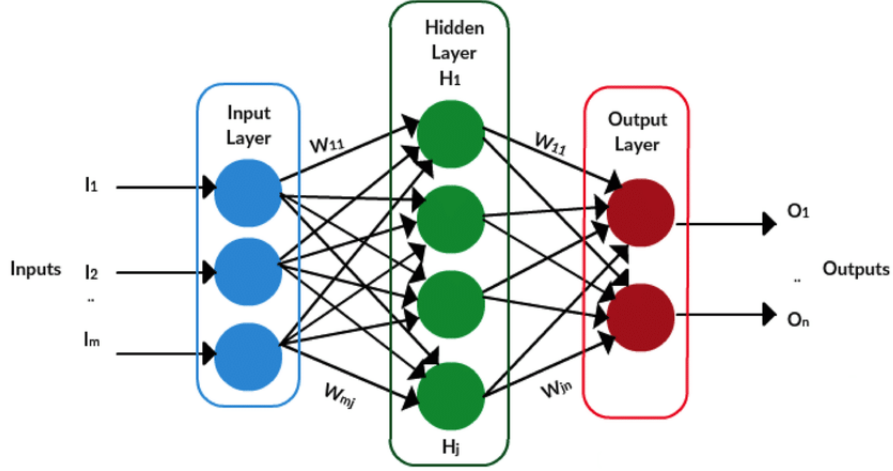


Figure 3.9: The MLP architecture with m inputs, one hidden layer and n outputs [42].

The outputs of an MLP are calculated based on the inputs, weights, and biases as follows:

- 1) First the weighted sums of input values are determined using the Equation 3.2

$$t_j = \sum_{i=1}^n (W_{i,j}, x_i) - B_j, \quad j = 1, 2, \dots, h \quad (3.2)$$

where n is the number of input neurons, $W_{i,j}$ denotes the link weight from the i^{th} neuron in the input layer (x_i) to the j^{th} neuron in the hidden layer (h_j), (x_i) shows the i^{th} input and B_j symbolizes the bias of the j^{th} hidden node.

- 2) In the second step, by using an activation function, the output value of every neuron in the hidden layer is computed as follows in Equation 3.3:

$$T_j = \text{sigmoid}(t_j) = \frac{1}{(1 + \exp(-t_j))}, \quad j = 1, 2, \dots, h \quad (3.3)$$

- 3) The final output of the network is described depending on the outputs of the hidden nodes as below in Equation 3.4, 3.5:

$$o_k = \sum_{j=1}^h (W_{j,k}, T_j) - B'_k, \quad k = 1, 2, \dots, m \quad (3.4)$$

$$O_k = \text{sigmoid}(o_k) = \frac{1}{(1 + \exp(-o_k))}, \quad k = 1, 2, \dots, m \quad (3.5)$$

Where W_{jk} is the correlation weight between the j^{th} hidden neuron and the k^{th} output neuron. B'_k is the bias of the k^{th} hidden neuron [42].

In summary, MLP can learn complex functions through a series of layers and activation functions. It is trained using supervised learning algorithms and can be optimized using various optimization techniques. Careful selection of hyperparameters is necessary to achieve optimal performance.

3.3.3 AlexNet Classifier Architecture

AlexNet, an influential deep convolutional neural network (CNN) devised by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012, has earned wide recognition as a pioneering model that revolutionized the field of computer vision. It has set new benchmarks in image classification tasks, exemplified by its victory in the prestigious ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012, a landmark competition in the realm of computer vision [43]. While AlexNet was initially designed for image classification, its architecture can also be adapted for other computer vision tasks, such as object detection or image segmentation, by modifying or extending its layers and incorporating additional techniques. However, at its core, It is a powerful image classifier.

A defining feature of AlexNet is its deep architecture, comprising eight layers that encompassed five convolutional and three fully connected layers. This innovative structure enabled AlexNet to capture complex features from raw image data, including low-level details such as edges and corners and high-level semantic features such as object parts and textures. By leveraging this depth, it achieved unparalleled accuracy in image classification tasks, surpassing earlier models that employed shallower architectures [44].

AlexNet made another significant contribution to the field of deep learning by introducing Rectified Linear Units (ReLU) as activation functions. This innovation addressed the notorious vanishing gradient problem often encountered in deep neural networks. This problem arises when gradients diminish significantly during backpropagation, resulting in sluggish convergence and subpar performance. By incorporating ReLU activations, AlexNet effectively mitigated this issue, facilitating faster and more efficient training of the network. Consequently, ReLU activations enabled it to acquire enhanced representations of image data, leading to improved classification results.

On the other hand, AlexNet revolutionized the field of deep learning by introducing dropout regularization as a powerful technique to prevent overfitting during training. Overfitting, which happens when a neural network becomes overly specialized to the training data and struggles to generalize to unseen data, is a common challenge in machine learning. Dropout regularization addresses this issue by randomly setting a fraction of neurons to zero during each training iteration. This forces the network to rely

on different sets of neurons for different examples, preventing over-reliance on specific neurons and promoting more robust generalization. The breakthrough use of dropout regularization in AlexNet significantly improved its generalization performance, effectively mitigating overfitting and enhancing its model's ability to handle real-world data. The basic architecture of the AlexNet classifier is illustrated in Figure 3.10 [45].

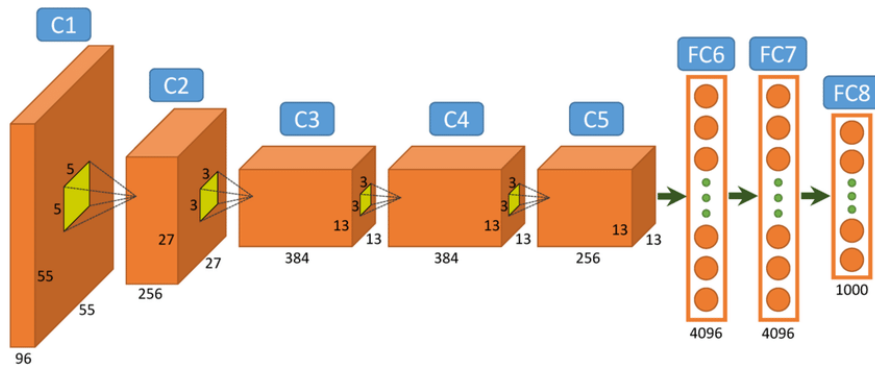


Figure 3.10: Architecture of Alexnet. From left to right (input to output) five convolutional layers with Max Pooling after layers 1,2, and 5, followed by a three-layer fully connected classifier (layers 6-8). The number of neurons in the output layer is equal to the designed number of output classes [45].

AlexNet's network structure is composed of 8 layers, which consist of 5 convolutional layers, 2 fully connected layers, and a softmax output layer. A summary of the architecture is presented below at a conceptual level:

- **Input Layer:** AlexNet takes an input image of size $227 \times 227 \times 3$ (where 3 is the number of colour channels - red, green, and blue).
- **Convolutional Layer 1:** This layer consists of 96 filters of size $11 \times 11 \times 3$ that are applied with a stride of 4, producing 96 feature maps of size $55 \times 55 \times 96$. A rectified linear unit (ReLU) activation function is applied to the output of each filter.
- **Max Pooling Layer 1:** A max pooling operation is applied to each of the 96 feature maps produced by the first convolutional layer, resulting in 96 feature maps of size $27 \times 27 \times 96$.
- **Convolutional Layer 2:** This layer consists of 256 filters of size $5 \times 5 \times 48$ (where 48 is the number of feature maps produced by the first convolutional layer). The filters are applied with a stride of 1, producing 256 feature maps of size $27 \times 27 \times 256$. ReLU activation is again applied to each output.
- **Max Pooling Layer 2:** A max pooling operation is applied to each of the 256 feature maps produced by the second convolutional layer, resulting in 256 feature maps of size $13 \times 13 \times 256$.

- **Convolutional Layer 3:** This layer consists of 384 filters of size $3 \times 3 \times 256$. The filters are applied with a stride of 1, producing 384 feature maps of size $13 \times 13 \times 384$. ReLU activation is applied to each output.
- **Convolutional Layer 4:** consists of 384 filters of size $3 \times 3 \times 192$ (where 192 is the number of feature maps produced by the previous layer). The filters are applied with a stride of 1, producing 384 feature maps of size $13 \times 13 \times 384$. ReLU activation is applied to each output.
- **Convolutional Layer 5:** This layer consists of 256 filters of size $3 \times 3 \times 192$. The filters are applied with a stride of 1, producing 256 feature maps of size $13 \times 13 \times 256$. ReLU activation is applied to each output.
- **Max Pooling Layer 3:** A max pooling operation is applied to each of the 256 feature maps produced by the fifth convolutional layer, resulting in 256 feature maps of size $6 \times 6 \times 256$.
- **Flatten Layer:** The output of the last max pooling layer is flattened into a 1D vector of length 9216.
- **Fully Connected Layer 1:** This layer consists of 4096 neurons and is fully connected to the flattened output of the previous layer. ReLU activation is applied to each neuron.
- **Fully Connected Layer 2:** This layer consists of 4096 neurons and is fully connected to the previous layer's output. ReLU activation is applied to each neuron.
- **Output Layer:** The final layer of AlexNet is a softmax layer that outputs a probability distribution over the possible classes of the input image. For the ImageNet dataset, there are 1000 possible classes, so the output layer consists of 1000 neurons.

Moreover, the training process of AlexNet involves optimizing the network's parameters (weights and biases) based on a loss function, typically cross-entropy loss, which measures the dissimilarity between the predicted class probabilities and the ground truth labels. This process enables the network to learn to make accurate predictions and generalize its knowledge to unseen images during the testing or inference phase [45].

Chapter 4

Methodology

The previous chapter serves as a crucial cornerstone in this research, offering a comprehensive analysis and synthesis of the current state of technologies. Through an extensive study of numerous research papers, a thorough survey was conducted to identify the most suitable methods for the development of a new face recognition system. This investigation revealed multiple approaches that can be utilized, and after careful consideration, a combination of knowledge-based and image-based methods was selected for the face detection component. Additionally, a neural network approach was chosen for the face recognition part. These decisions were primarily driven by their seamless applicability and high reliability in achieving accurate and efficient face detection and recognition.

In this section, the focus shifts to presenting the dataset and methodologies employed in this study. The chosen dataset provides a comprehensive collection of face images, carefully curated to represent a diverse range of individuals and pose variations. This dataset acts as a foundation for evaluating and refining the proposed methodologies. To enhance the accuracy and performance of the face recognition system, a range of techniques and algorithms have been implemented. These include preprocessing steps to improve the quality of the input images, feature extraction methods to capture discriminative facial characteristics, and advanced classification algorithms to accurately identify individuals. The utilization of these techniques aims to optimize the recognition results and ensure robust performance in various scenarios.

Figure 4.1 provides an insightful overview of the key steps and phases involved in the proposed method. It illustrates the sequential flow of the system, highlighting the crucial stages of face detection, feature extraction, and classification. This visual representation aids in understanding the overall structure and progression of the developed system. By presenting the dataset and methodologies utilized, this section establishes a solid foundation for the subsequent analysis. The integration of various techniques and

algorithms enables the system to deliver enhanced face recognition capabilities, ultimately contributing to advancements in the field of biometric identification.

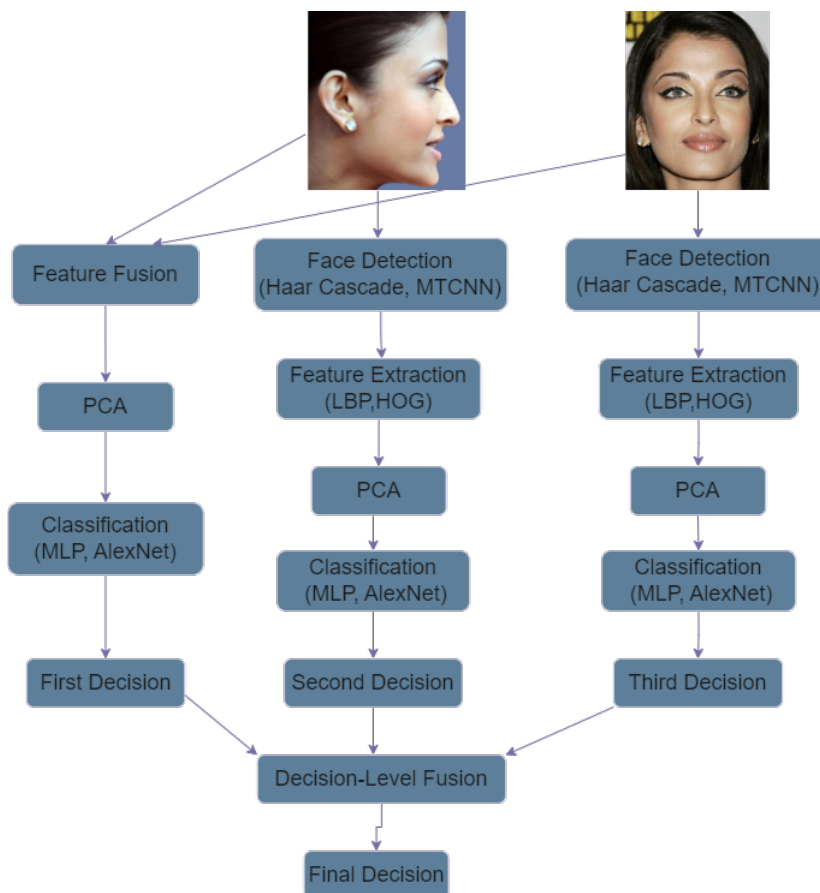


Figure 4.1: The Workflow of Proposed Face Recognition System

4.1 Dataset

The Celebrities in Frontal-Profile in the Wild (CFPW) data set used in this study is a collection of face images especially assembled to investigate the challenges associated with utilizing unconstrained face recognition models. The data collection is essentially a combination of constrained and unconstrained conditions with different ages of the same individual. The pictures are taken from freely available internet platforms but have been edited to meet particular "frontal" and "profile" postures. The fact that all other variants are unrestricted, allows us to explore the issue of pose variation in a more controlled manner [2].

The data set comprises a total of 7,000 images, featuring 500 individuals. For each individual, there are 10 frontal photographs and 4 profile photographs available. The original photos within the data set exhibit significant variation in terms of pixel size, with certain profile photographs lacking an ear or not being rotated by 90 degrees. To ensure a balanced data set, we constructed an evaluation subset by selecting samples from 140 individuals out of the total 500. In order to enhance performance, only a specific number of frontal and profile photographs per participant are chosen for this experiment. To accomplish this, we establish a threshold based on image size, specifically opting for pictures with dimensions equal to or greater than 150x200 pixels (height x width). Following dimension-based filtering, four frontal face images and four profile photographs are selected for each individual. Figure 4.2 provides a visual representation of some image samples from the CFPW data set [3].

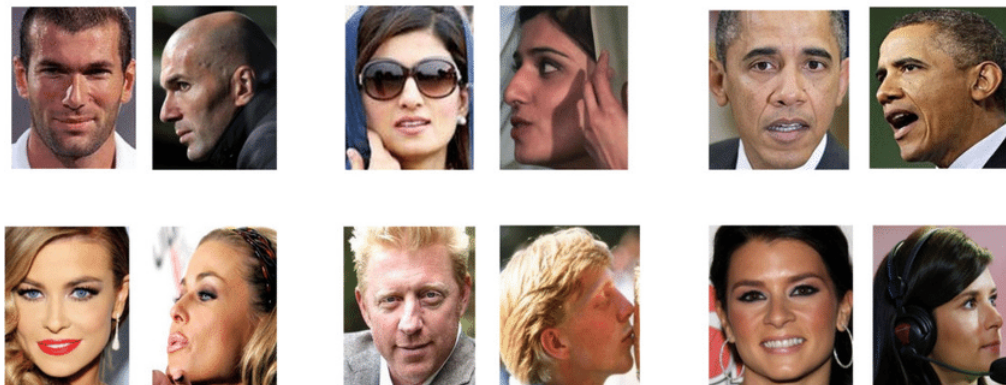


Figure 4.2: Sample Images from Celebrities in Frontal-Profile in the Wild (CFPW) data set [3].

4.2 pre-processing

Pre-processing is crucial in computer vision to handle images and produce the optimal image size and resolution. The original pictures in the dataset may contain unwanted noise and various light conditions, with different dimensions and colours. The pre-processing technique decreases probable noise and converts the image into a unique space thereby enabling effective classification and facilitating subsequent steps through the extraction of relevant features. By applying pre-processing procedures to the images, the CNN approach benefits from enhanced reliability and accelerated performance. These pre-processing techniques serve to eliminate distractions and enhance the overall quality of the input data, allowing CNN to focus on extracting meaningful patterns and features

that are crucial for accurate analysis and classification tasks [46]. As a result, CNN can operate with greater efficiency and effectiveness, leading to improved performance in terms of both reliability and speed. The fundamental image pre-processing steps are illustrated in Figure 4.3.

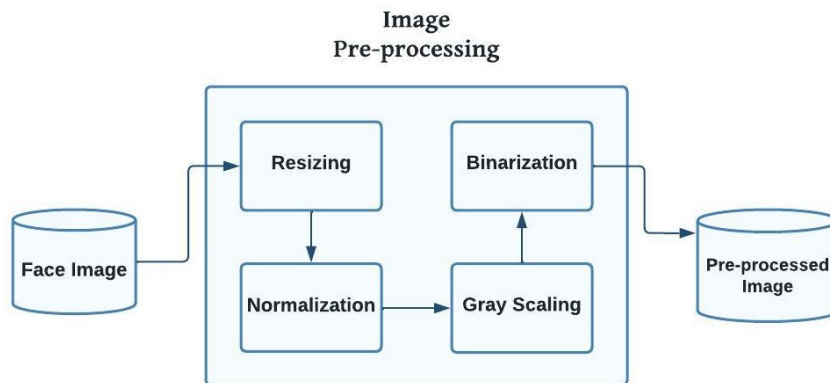


Figure 4.3: Block diagram of image pre-processing.

The following pre-processing methods are applied to transform the images:

- Normalization:** It is a crucial step in image processing, especially when dealing with photos captured under uncontrolled illumination conditions, as the distribution of concentration levels may vary significantly. To address this challenge, a series of procedures were employed to ensure consistent levels of image intensity throughout the dataset. By applying normalization techniques, variations in lighting conditions are effectively mitigated, resulting in images that exhibit more uniform and balanced intensity levels. This process aids in reducing the influence of external factors such as lighting variations, shadows, and highlights, which can hinder accurate face recognition. The resulting normalized facial images possess enhanced clarity and consistency, which is advantageous for subsequent analysis and recognition tasks. The normalization process facilitates the extraction of discriminative facial features and contributes to achieving a higher recognition rate by reducing the impact of inconsistent illumination conditions on the overall performance of the recognition system.
- Binarization:** The binarization method is used to separate the white and black pixels of an image by applying thresholding to the pixel values. It can be categorized as either global or local, depending on the thresholding approach employed. Global thresholding involves selecting an intensity value as the threshold for distinguishing between white and black in the entire image. On the other hand, local thresholding

divides the image into overlapping regions and applies partial segmentation methods within each region. Global binarization may not be suitable for images with noise or texture since it does not account for these factors. In such cases, it is recommended to use adaptive thresholding, which adjusts the threshold value dynamically to accommodate variations in the image.

- **Resizing:** In computer vision, image resizing is a crucial pre-processing step that plays a significant role in optimizing various tasks. By resizing images, we effectively eliminate unnecessary elements and focus on the essential content. This technique not only helps conserve memory resources but also significantly improves computational speed. Moreover, resizing images to smaller dimensions is particularly beneficial when working with deep learning models, as they tend to train more efficiently on smaller images. Thus, by resizing images, we can expedite the training process while still capturing the critical information needed for accurate analysis.
- **Gray Scaling:** It is a procedure for transforming images where the pixel values are determined by the brightness of the image. Since processing colored images can be challenging for a CNN architecture, grayscale conversion is often employed. By converting images to grayscale, facial recognition using CNN can be effectively performed without relying on color information from the input picture. This approach ensures that facial features can be accurately detected and recognized, regardless of color variations.

4.3 Implementation

This study focuses on applying face recognition models to the CFPW dataset, which comprises frontal and profile face images. The model will take input data in the form of 128x128 pixel images and generate output classes corresponding to each celebrity. The dataset is organized into folders, with each folder containing two sub-folders: frontal and profile. As previously mentioned, the dataset includes 7000 images from 500 individuals, with each user having 10 frontal and 4 profile photographs. The original photos in the dataset vary in terms of pixel size significantly. Thus, to ensure a well-balanced dataset for our evaluation subset, we carefully selected samples from 140 individuals out of the total 500. In order to enhance performance, we specifically choose a limited number of frontal and profile photographs from each participant for this experiment. This selection process involves establishing a threshold based on image size, where we opt for pictures with dimensions equal to or greater than 150x200 pixels (height x width). After applying

this dimension-based filtering, we ultimately select four frontal face images and four profile photographs for each person.

Initially, the data will undergo pre-processing, and the face detection method will be employed to extract the region of interest. Subsequently, the accuracy of the models will be assessed by applying the classifier to the cropped photos and extracting features from each image independently. The implementation of the project is divided into the following five distinct phases:

4.3.1 Data Pre-processing

The pre-processing phase in face recognition involves several steps to ensure the data is appropriately prepared for subsequent analysis. In this study, the implementation of the pre-processing stage includes resizing, grayscale conversion, normalization and binarization. By implementing these pre-processing steps, the face recognition system can effectively prepare the images for subsequent feature extraction and classification stages, leading to more accurate and reliable recognition results.

As previously stated, the dataset comprises images that display considerable diversity in terms of their size and dimensions, presenting obstacles to maintaining consistency during analysis and processing. To address this issue, a crucial step is to resize the images to a standardized dimension, ensuring uniformity throughout the dataset. In this particular study, the images are resized to a resolution of 128x128 pixels, enabling effective comparability and facilitating subsequent computational operations. Images can be binarized and normalized before or after the face detection phase. In the case of grayscale images, normalization involves subtracting the minimum value of the image data from the grayscale values and subsequently dividing it by the maximum value of the grayscale image. This process ensures that the grayscale data is scaled and adjusted to a standardized range, facilitating further analysis and comparison.

4.3.2 Face Detection

Once the images have undergone pre-processing, the next step is to extract the region of interest, which specifically refers to the face in this study. In this regard, two face detection methods have been employed: the Haar Cascade and the MTCNN. The Haar Cascade algorithm is rooted in the concept of Haar-like features, utilizing a machine learning approach to detect patterns that signify facial features. This algorithm has proven to be effective in identifying facial characteristics by leveraging learned patterns and employing classification techniques.

The Haar Cascade method effectively filters out irrelevant features from images, enabling the identification of distinct human faces. This method employs two essential parameters: the scale factor and `min_neighbors`. The scale factor determines the extent to which the image size should be reduced during detection. We have chosen a scale factor of 1.1, striking a balance between resizing rate and detection speed. This ensures that the reduction in size is not too significant, allowing for efficient detection.

On the other hand, `min_neighbors` define the number of neighbors required in each rectangle for a detection to be considered valid. A higher value may result in more accurate detection but can also lead to a higher chance of missing certain faces. For our experiment, we have set the `min_neighbors` to 9, ensuring a reasonable balance between precision and avoiding false negatives. Additional parameters, such as `minsize` and `maxsize`, remain unchanged in this context and are not explicitly modified for our experiment. Figure 4.4 shows an example of an extracted face using the Haar Cascade algorithm. [fig a] demonstrate the original image and [fig b] depicts cropped image which solely focuses on the extracted face itself. Once the Haar Cascade algorithm determines the facial region, we extract this area by cropping it out from the original image and discarding the surrounding context and unrelated details.

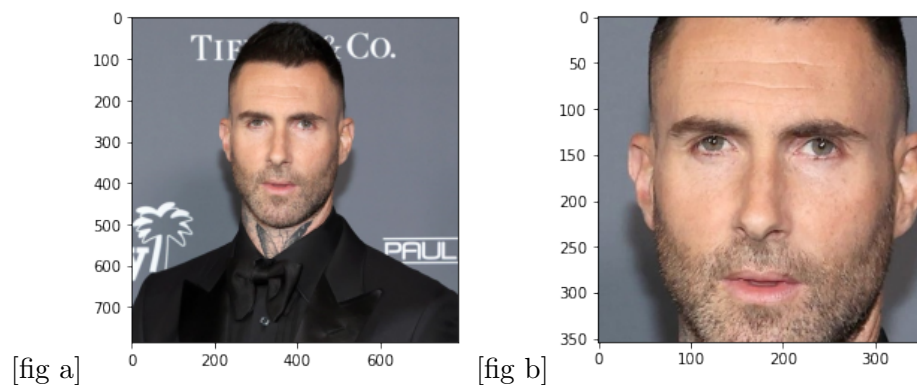


Figure 4.4: sample of original and extracted faces

After applying the Haar cascade classifier, we employ the MTCNN algorithm to identify facial features such as lips, eyes, and nose in the given images. Once the MTCNN process accurately determines the bounding boxes for faces, we represent them as rectangular shapes. It enables us to extract valuable information about the precise locations and shapes of facial features. Figure 4.5 indicates marked facial features and bounding boxes using MTCNN.

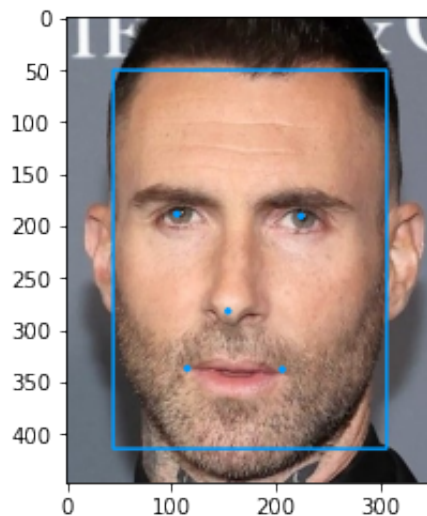


Figure 4.5: Finding facial landmarks using MTCNN

In summary, the combined power of the Haar Cascade classifier and MTCNN elevates our facial analysis capabilities. The precise detection of facial landmarks through MTCNN enhances our ability to extract meaningful insights from facial images, facilitating a deeper understanding of facial features and their spatial relationships. These valuable advancements allow for more accurate and reliable facial analysis, ultimately contributing to a wide range of applications in various domains.

4.3.3 Feature Extraction

As mentioned earlier, features in images are significant local intensity variations resulting from shifts in pixel values across an image. Notably, edges and corners can be deemed as crucial features within an image. Employing edge detection and feature descriptor techniques proves beneficial in identifying image features that enhance the performance of face recognition. In this study, we have utilized robust feature detectors and extractors on the datasets, facilitating effective analysis.

The HOG technique leverages the magnitude and angle of gradients within a given region to construct informative histograms. Several essential parameters shape the HOG algorithm: orientation, determining the number of orientation bins per cell; pixel per cell, defining the cell size in pixels; and cell per block, specifying the block configuration. In this context, we utilize 9, (8,8), and (2,2) as orientation, pixel-per-cell, and cell-per-block values, respectively. These parameters are implemented using the Skimage Python library, which offers effective methods for HOG analysis. In the subsequent phase, we employ the LBP algorithm to extract the most prominent features from the data. One of the

primary advantages of LBP lies in its robustness to rotation and grey scale variations. LBP serves as a fundamental technique for identifying distinctive characteristics within an image, while effectively adapting to changes in lighting conditions. This method incorporates two crucial parameters: "p" and "r." The former denotes the number of points forming the circular neighbourhood, while the latter represents the radius of the neighbourhood circle. Various parameter values have been explored; however, we have found that for our specific dataset, the most suitable choices are 2 for "p" and 8 for "r". It returns the resulting LBP image, where each pixel contains the binary code computed using the LBP algorithm. The LBP algorithm assumes a grayscale input image. To ensure compatibility, we can leverage the capabilities of the OpenCV library to convert the image from BGR to grayscale. Additionally, it is important to note that all input images provided to the LBP function must have the same size. Consequently, we resized the images to a standardized dimension of 128x128 pixels.

To enhance the feature representation, we perform feature fusion by merging and normalizing the results obtained from LBP and HOG. This fusion process yields combined features that capture the complementary strengths of both techniques. The Canny edge detection technique has been employed on the project dataset to enhance the visibility of image edges. This method proves highly effective in detecting prominent features within an image. However, due to the susceptibility of Canny methods to noise, the initial step involves a lowpass filtering process, such as Gaussian blur, to minimize image noise. Subsequently, the derivative is computed along the horizontal (x-axis) and vertical (y-axis) directions by convolving the image with Sobel x and y operators, respectively. This computation allows for the retrieval of both the magnitude and phase of the image. Furthermore, the obtained edges undergo maximum suppression to reduce their thickness.

The Canny edge detection function, available in the Python library, encompasses several adjustable parameters that influence the segmentation process. These parameters include a lower threshold, a higher threshold, and a second gradient. Setting appropriate threshold values is crucial for effective segmentation, while the second gradient parameter determines the level of edge detection within the image. In our study, we have chosen a lower threshold of 100, and a higher threshold of 200, and utilized the second gradient. The result of the feature extraction and feature fusion algorithms is illustrated in figure [4.6](#).

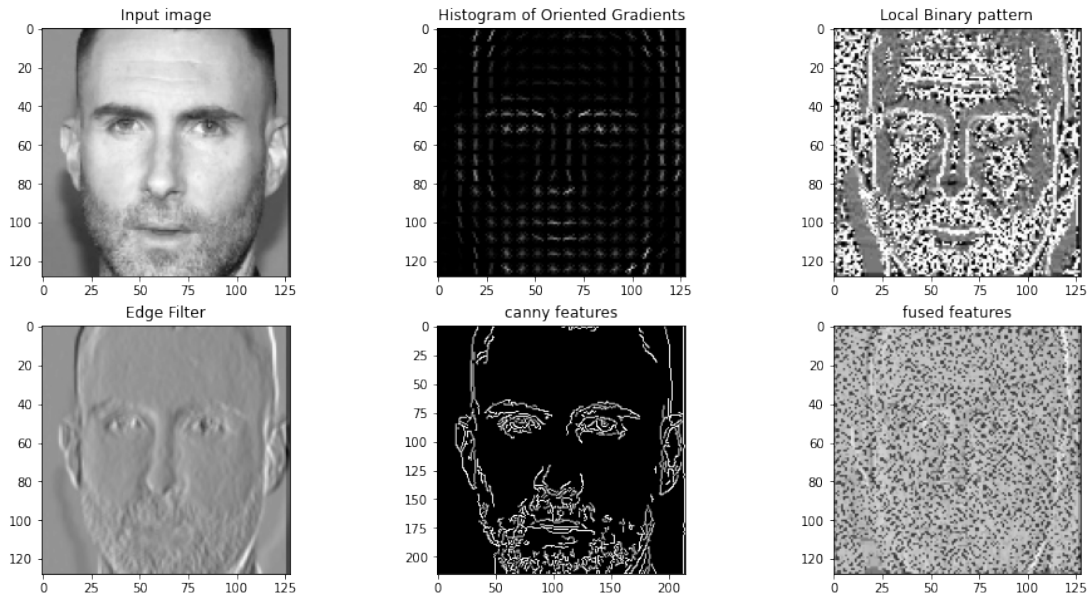


Figure 4.6: Feature Extraction and Feature Fusion

4.3.4 Split Data into Train and Test Subsets

The dataset consists of 8 face images per subject, comprising four frontal faces and four profile faces. To ensure equitable distribution, 80 percent of the face images will be utilized for training, with the remaining 20 percent allocated for testing. To maintain balance, the stratify feature will be employed, ensuring an equal number of training and test images for each subject.

As previously mentioned, the experiment involves three distinct image subsets. One subset exclusively contains frontal faces, another consists solely of profile faces, and the third subset combines both frontal and profile faces. Consequently, for each subject, there will be 3 training images and 1 test image in separate subsets for frontal and profile faces. Additionally, the third subset, comprising merged frontal and profile faces, will have 6 training samples and 2 testing images. Table 4.1 shows the distribution of data for the train and test set for each subset.

After splitting data into train and test subsets we normalize the images in each subset. To standardize the features of a dataset, we can utilize the **StandardScaler()** function from the sklearn library in Python. By creating an instance of the StandardScaler class, we can effectively scale the features of our dataset. The **fit_transform()** method is applied to the training data (X_{train}), which calculates the mean and standard deviation of the features and transforms the data accordingly. On the other hand, the transform method is used to apply the same scaling transformation to the testing data (X_{test})

Table 4.1: Distribution of data for train and test subsets.

Class Name	#of People	#of Images	#of Images for training	#of Images for testing
Frontal	140	560	420	140
Profile	140	560	420	140
Frontal-Profile	140	1120	840	280

based on the parameters learned from the training data. This ensures that the training and testing datasets are scaled consistently.

4.3.5 Dimensionality Reduction using PCA algorithm

As previously stated, the PCA algorithm is widely utilized in face recognition applications for its effectiveness in dimensionality reduction. It aims to extract the most informative features from high-dimensional data while minimizing the loss of relevant information. In the context of face recognition, it can significantly reduce the dimensionality of face images by projecting them onto a lower-dimensional subspace.

It is highly suitable for face recognition applications due to its exceptional capabilities in dimensionality reduction, face representation, discriminative power, and robustness to variations. By reducing the dimensionality of face images, PCA significantly reduces computational complexity and memory requirements while preserving essential facial features. Through the identification of eigenfaces, which represent the principal components of face data, it captures the most significant variations in face images, providing a compact and efficient representation for face recognition.

Moreover, PCA's focus on capturing maximum variance enables the identification of discriminative facial features, allowing for accurate matching and recognition. Additionally, PCA exhibits robustness to variations in lighting conditions, pose, and facial expressions, making it reliable in recognizing faces under diverse circumstances. Thus, it stands as an ideal choice for face recognition applications, offering unparalleled performance in multiple critical aspects.

Machine learning and deep learning models are specifically designed to operate on vectors. When working with image data, which is typically represented in matrix form, it is necessary to convert it into a vector format. Thus, before applying the PCA algorithm, we reshape the images into a 2D vector representation.

4.3.5.1 PCA Projection

To implement the PCA algorithm, initially, we import the PCA class from the scikit-learn library, which is used for performing dimensionality reduction. Then an instance of the PCA class is created, specifying that we want to reduce the dimensionality of the data to two components. Later the PCA model is fitted to the input data. This step calculates the principal components based on the covariance matrix of the data and determines the transformation matrix that maps the original data to the reduced-dimensional space. The fitted PCA model is then used to transform the input data into the reduced-dimensional space. The result contains the transformed data, where each sample is represented by its corresponding two principal components. The scatter plot for the PCA Projection of 20 people is illustrated in figure 4.7:

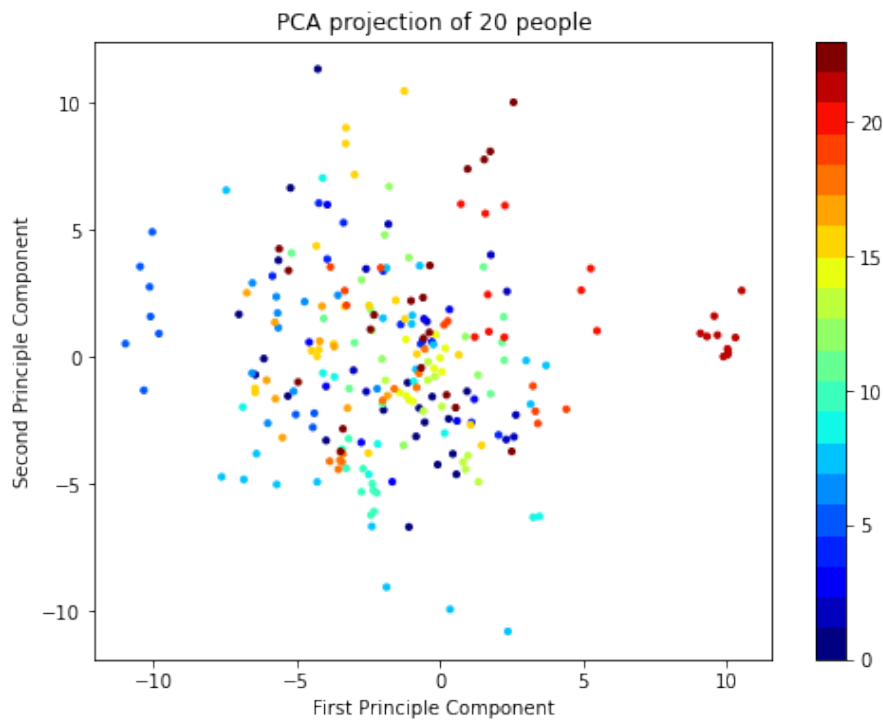


Figure 4.7: PCA Projection of 20 people

4.3.5.2 Finding Optimum Number of Principal Component

In this section, An instance of the PCA class is created without specifying the number of components. By default, the algorithm will compute as many components as there are features in the input data. Then, the PCA model is fitted to the input data. This step calculates the principal components based on the covariance matrix of the data

and determines the transformation matrix. After creating an instance of the PCA class, `pca.fit(x_train)` fits the PCA model to the training data (`x_train`), calculating the principal components and their corresponding explained variances. The result is displayed by the plot using the matplotlib library in Python. The plot shows a line graph where the x-axis represents the components (usually ordered from 1 to the number of features in the `x_train`), and the y-axis represents the corresponding explained variances. The graph helps visualize how much variance in the original data is explained by each principal component as shown in figure 4.8:

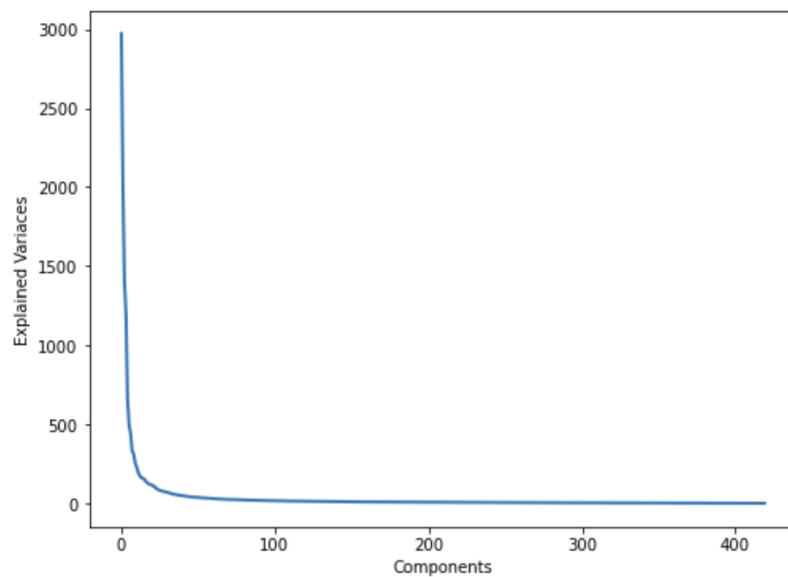


Figure 4.8: Finding Optimum Number of Principal Component

In the figure above, we observe that using 90 or more Principal components results in representing the same data. This indicates that the majority of the data's variance can be captured using these 90 components. Thus we proceed with the classification process by utilizing these 90 PCA components, reducing the data's dimensionality while maintaining the essential information. The next step is to generate the average faces and eigenfaces from the PCA analysis. The average face helps us understand the common characteristics and serves as a reference for comparison, while eigenfaces enable dimensionality reduction, feature extraction, and face reconstruction. Then we perform dimensionality reduction on the training and test data. Using the PCA model, the dimensionality reduction transformation applies to the training data (`x_train`). The `transform()` method is used to project the data onto the principal components obtained during the training phase. This step reduces the dimensionality of the training data from the original feature space to the space defined by the principal components. The same dimensionality reduction

transformation is performed using the PCA model on the test data (x_{test}). The test data is projected onto the same principal components obtained from the training data. This ensures that the dimensionality reduction is consistent between the training and test datasets, allowing for proper evaluation and comparison. The reduced-dimensional data will be used as input for machine learning models or other analysis techniques to benefit from lower-dimensional representations of the data.

4.3.6 Classification

After performing image pre-processing and extracting crucial features, the processed data can be inputted into classifiers. Given our objective of distinguishing individual faces within the dataset, our model must serve as a classifier capable of labeling the results. In this study, we utilize two powerful deep learning classifiers: the MLP and AlexNet, Both models have been pre-trained on extensive datasets, such as ImageNet, and exhibit CNN architectures, rendering them well-suited for the task at hand. Before building the model we need to convert categorical labels into one-hot encoded vectors. we use `to_categorical()` function from the Keras.utils module which is widely used in multi-class classification tasks to convert the target variables into one-hot encoded format. This format is commonly used in deep learning frameworks, including Keras, as it provides a suitable representation for training models to predict categorical variables with multiple classes.

The accuracy and performance of a deep learning model are greatly influenced by the choice of various method parameters. By carefully defining the appropriate loss function, activation function, learning rate, and number of hidden nodes, we can effectively impact the convergence and overall performance of the model. For instance, utilizing the softmax function in the last layer enables the generation of probabilities for all potential classes, ensuring that their summation equates to one. Consequently, the class with the highest probability is identified as the model's output, enhancing the interpretability and reliability of the model's predictions. To construct the MLP model, we employ a series of steps involving multiple dense layers, dropout regularization, and a softmax output layer. Subsequently, the model is compiled, incorporating the desired optimizer, loss function, and evaluation metrics outlined below:

- we initialize a sequential model using the `Sequential()` function from Keras. This allows us to build a neural network model in a sequential manner, where each layer is added one after another.

- The `add()` method is used to add a Dense layer to the model. The first Dense layer has 256 units and uses the ReLU activation function. It also specifies `input_dim=90`, indicating that the expected input shape for this layer is 90 features.
- A Dropout layer is added after the first Dense layer with a dropout rate of 0.2. Dropout helps prevent overfitting by randomly setting a fraction of the input units to 0 during training, reducing interdependencies between neurons.
- Additional Dense and Dropout layers are added to the model, gradually reducing the number of units. This pattern of adding layers with decreasing number of units is a common approach in deep learning architectures.
- The final Dense layer is added with 140 units, representing the number of classes in the multi-class classification task. It uses the softmax activation function to output the probability distribution over the classes.
- The number of training epochs is set to 150, indicating how many times the model will iterate over the training data during training.
- The batch size is set to 128, which determines the number of samples that will be propagated through the network at once.
- The `ReduceLROnPlateau()` callback is created, which monitors the validation accuracy. If no improvement is observed in the validation accuracy for 2 epochs (`patience=2`), the learning rate is reduced by a factor of 0.1 (`factor=0.1`).
- The model is compiled with the Adam optimizer, which adapts the learning rate during training. The learning rate is set to 0.001 (`lr=1e-3`).
- The loss function is set to `'categorical_crossentropy'`, which is commonly used for multi-class classification tasks. The model will also compute and report the accuracy metric during training.
- Finally, we provide a summary of the model's architecture including the layer type, output shape, and the number of parameters in each layer.

The next step is to implement the AlexNet model for face classification in Python using the TensorFlow library. After importing the necessary libraries, We define the `alexnet_model` function, which takes the input shape and number of classes as arguments and returns the AlexNet model. Inside the model function, we create a sequential model using `tf.keras.models.Sequential()`. We add the layers of the AlexNet model one by one using `model.add()`. Each layer is instantiated with specific parameters such as the number of filters, kernel size, strides, padding, and activation function. The

model starts with two convolutional layers followed by max-pooling layers (layers 1-4). The next three convolutional layers (layers 5-7) do not have pooling layers in between. The model continues with another max pooling layer (layer 8) and then a flatten layer (layer 9) to convert the output from the previous layers into a 1D feature vector. Two fully-connected layers with ReLU activation are added (layers 10 and 12). Dropout layers with a regularization rate of 0.5 are included after each fully-connected layer to prevent overfitting (layers 11 and 13). Finally, an output layer with softmax activation is added to classify the input into the specified number of classes (layer 14). The `alexnet_model` function returns the constructed model. We define the `input_shape` (128x128x3) and the `num_classes` (140 in this case) for our specific classification task. Then the instance of the AlexNet model is created by calling `alexnet_model(input_shape, num_classes)`. We compile the model using the Adam optimizer, sparse categorical cross-entropy loss function, and accuracy as the evaluation metric. The model summary is printed using `model.summary()`, which provides a detailed overview of the model architecture, including the shape and number of parameters in each layer.

The specific choices made in the model architecture of AlexNet, such as the number of filters, kernel sizes, and activation functions, are crucial for achieving effective face classification. we discuss these choices and their relevance to the task of face classification below:

- **Number of Filters**

The number of filters employed in the convolutional layers plays a vital role in capturing distinctive features from face images. By increasing the number of filters, the model becomes capable of extracting a wider range of complex and diverse features. AlexNet implements a progressive increase in the number of filters in deeper layers. This approach facilitates the acquisition of hierarchical features, starting from 96 filters in the first layer, 256 in the third layer, 384 in the fifth and sixth layers, and finally 256 in the seventh layer. This hierarchical progression enables the model to learn a spectrum of features, ranging from low-level edges to high-level object parts and structures, ultimately enhancing face classification performance.

- **Kernel Sizes**

The selection of kernel sizes profoundly impacts the receptive field of each convolutional layer and influences the size of the learned features. Smaller kernel sizes focus on capturing intricate details, while larger kernel sizes emphasize global structural elements. In this model, larger kernel sizes are employed in the earlier layers (11x11 in the first layer and 5x5 in the third layer) to capture broader patterns and edges

that provide a foundation for subsequent feature extraction. Conversely, smaller kernel sizes (3x3) are employed in the deeper layers to extract more localized and fine-grained facial features, enabling a more comprehensive representation.

- **Activation Functions**

Activation functions introduce non-linearity to the model, enabling the exploration of complex relationships between features and enhancing the model's discriminative power. AlexNet leverages the Rectified Linear Unit (ReLU) activation function, which has demonstrated remarkable efficacy in training deep neural networks. ReLU effectively mitigates the vanishing gradient problem, expedites training convergence, and introduces sparsity within the network. These advantages enhance the model's capacity for generalization, reduce overfitting, and enable the acquisition of highly informative facial features during face classification.

Chapter 5

Experimental Evaluation

In this chapter, we will provide a detailed description of the experimental setup conducted to evaluate our proposed face recognition system. This encompasses the selection of evaluation metrics, the specific configurations of the algorithms used, and the relevant parameters employed. Additionally, we will present the outcomes obtained from these experiments. To begin with, the evaluation metrics chosen for assessing the performance of our face recognition system were primarily focused on accuracy rates. The accuracy metric provides an overall indication of the system's ability to correctly identify individuals. The experimental setup employed both unimodal and multimodal approaches for face recognition. The unimodal approach utilized the frontal face and profile face images separately, while the multimodal approach incorporated both frontal and profile face images. The inclusion of profile face images was motivated by the idea that acquiring features from different angles can enhance the system's ability to recognize faces accurately.

5.1 Experimental Setup

The dataset utilized in this study comprises 140 individuals sourced from the CFPW-Dataset. To ensure comprehensive coverage, four frontal face images and four profile images were selected for each person. These 140 individuals were carefully chosen from a larger pool of 500 individuals available in the CFP dataset. This selection was necessary due to the presence of numerous profile images lacking ear visibility or not being rotated by the standard 90° angle. To accurately identify and analyze the facial features, both frontal and profile faces were detected using the Haar cascade algorithm. To extract the most crucial facial characteristics, various feature extraction methods, such as LBP and HOG, were employed. Additionally, the PCA algorithm was utilized to reduce the image

dimensions. Ultimately, to identify the individuals in the dataset, two different CNN classifiers, namely MLP and AlexNet, were employed.

The reliability and efficiency of the proposed method are thoroughly assessed through extensive experimentation conducted on the dataset. After extracting the features from the frontal and profile face images, the next step involves combining them into a single sample. This fusion process aims to leverage the complementary information present in both the frontal and profile views of the same individual. By combining the features, the system can create a more comprehensive representation of the face, potentially improving recognition accuracy. Once the features from the frontal and profile faces are fused into a single sample, the system proceeds with decision-level fusion. In this stage, the final decision is made by consolidating the decisions obtained from the fused features. A majority voting scheme is employed, where each individual feature contributes to a vote, and the decision with the highest number of votes is considered a final decision. The decision-level fusion with majority voting helps mitigate the potential errors or uncertainties in individual feature-based decisions. By considering multiple perspectives and combining the decisions, the system aims to enhance the overall robustness and accuracy of the face recognition process.

5.2 Experimental Results

Among the unimodal systems as shown in 5.1, the MLP model achieved an accuracy of 81.50% for frontal face recognition, while the CNN model, specifically AlexNet, outperformed the MLP with an accuracy of 89.45% for frontal face recognition. This indicates the superior capability of CNN models in capturing intricate facial features and patterns, resulting in more accurate recognition. For profile face recognition, the MLP model achieved an accuracy of 78.56%, while the CNN (AlexNet) model obtained an accuracy of 80.34%. Although the CNN model's performance was slightly better, both models demonstrated reasonable accuracy rates for profile face recognition, which can be challenging due to the variation in facial features across different angles.

Moving on to the multimodal systems shown in 5.2, combining different modalities such as LBP, PCA, and MLP further improved the recognition accuracy. The combination of the LBP, PCA, and MLP approach yielded an accuracy of 84.13% for frontal face recognition and 82.04% for profile face recognition. This suggests that incorporating additional modalities can enhance the system's ability to capture diverse facial characteristics and improve its performance. Moreover, fusing features from both frontal and profile faces using the LBP, PCA, and MLP techniques resulted in an accuracy of 88.25%, indicating the benefits of utilizing multiple viewpoints for face recognition. This approach

demonstrates that considering facial features from different angles can lead to more robust and accurate recognition results. Additionally, combining the powerful AlexNet CNN model with LBP further improved the system’s performance. For frontal face recognition, the AlexNet + LBP approach achieved an accuracy of 90.83%, while for profile face recognition, the accuracy was 86.68%. The fusion of both frontal and profile faces using AlexNet and LBP resulted in an outstanding accuracy rate of 96.40%. These results emphasize the importance of leveraging advanced deep learning models and incorporating complementary modalities for achieving highly accurate face recognition. The recognition rates of faces under the identification mode of unimodal and multimodal systems with the proposed systems are shown in Table 5.1 and Table 5.2 for the experiments applied to CFPW dataset.

Table 5.1: Recognition Rate for Unimodal System.

	Algorithms	Traits	Recognition Rate (%)
Unimodal systems	MLP	Frontal Face	81.50
	CNN (AlexNet)	Frontal Face	89.45
	MLP	Profile Face	78.56
	CNN (AlexNet)	Profile Face	80.34

Table 5.2: Recognition Rate for Multimodal System.

	Algorithms	Traits	Recognition Rate (%)
Multimodal systems	LBP + PCA + MLP	Frontal Face	84.13
	LBP + PCA + MLP	Profile Face	82.04
	LBP + PCA + MLP	Frontal + Profile	88.25
	AlexNet + LBP	Frontal Face	90.83
	AlexNet + LBP	Profile Face	86.68
	AlexNet + LBP	Frontal + Profile Face	96.40

Overall, the findings highlight the significance of employing multimodal approaches and integrating various techniques such as deep learning models, feature fusion, and different modalities to enhance the performance of face recognition systems. The multimodal systems consistently outperformed their unimodal counterparts, demonstrating the effectiveness of leveraging multiple sources of information. The high accuracy rates achieved by these systems provide promising prospects for real-world applications, including

security systems, access control, and identity verification, where accurate and reliable face recognition is crucial.

The analysis of the learning curve for model loss in AlexNet has been illustrated in Figure 5.1 which involves studying the decrease in loss function value over multiple training iterations or epochs. The learning curve provides insights into how the model's loss decreases and converges as it learns from the training data. As it is shown, there will be no significant improvement in the loss value of the model. Hence we decided to stop at epoch number 50.

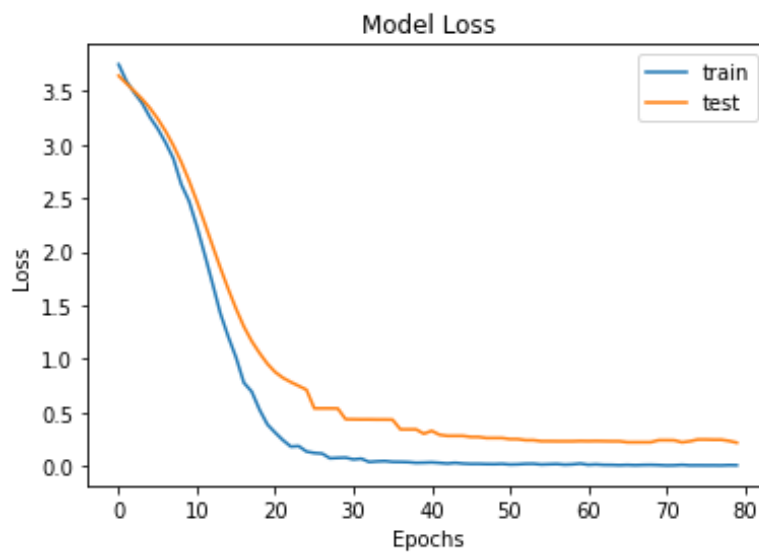


Figure 5.1: AlexNet Model Loss

Chapter 6

Conclusions

In this study, the extraction of valuable features plays a critical role in the development of an accurate face recognition system. To achieve this, the LBP approach is employed in conjunction with CNNs. This combined approach allows for the extraction of discriminative features from facial images, enabling effective facial representation.

Furthermore, the performance of the system is further enhanced through the utilization of feature-level and decision-level fusion techniques. Feature-level fusion combines the extracted features from multiple modalities, while decision-level fusion combines the classification decisions made by individual classifiers. These fusion techniques leverage the complementary information provided by different modalities, thereby improving the overall recognition accuracy.

To evaluate the system's performance, extensive experiments are conducted on the diverse CFPW dataset. This dataset includes frontal and profile face images captured under various conditions, mimicking real-world scenarios. The results of these experiments demonstrate that multimodal approaches outperform unimodal ones, highlighting the importance of considering multiple modalities for effective biometric recognition.

Remarkably, the fusion of frontal and profile images using the AlexNet model yields the highest accuracy rate of 96.40%. This outcome underscores the significance of incorporating multiple modalities, specifically frontal and profile images, to achieve robust and accurate face recognition. By combining these modalities, the system effectively mitigates the challenges posed by pose variation, resulting in improved recognition performance.

The findings from this study emphasize the effectiveness of multimodal approaches in biometric recognition systems. The utilization of multiple modalities, along with appropriate fusion techniques, enables the system to overcome the limitations associated

with pose variation and enhance the accuracy and reliability of face recognition. These insights contribute to the advancement of biometric recognition systems and pave the way for more robust and versatile applications in various domains.

The multimodal recognition systems employed in this study demonstrate a level of accuracy that is consistent with numerous other related studies. These studies have consistently shown that multimodal biometric systems surpass unimodal biometric systems. The reason behind this superiority lies in the abundance of biometric sources, which provide a wealth of information and discriminant features that are crucial for the recognition process. However, it is important to acknowledge the trade-off between accuracy and processing time when considering multimodal systems. These systems require additional time for certain recognition steps, such as acquiring data from multiple sources and performing fusion at different levels. While the benefits of increased accuracy are evident, it is crucial to carefully balance these advantages against the potential delays introduced by the additional processing requirements.

6.1 Future Directions

Future work can focus on several avenues to further improve and expand the capabilities of the multimodal face recognition system.

- **Integration of Additional Modalities:** While this study considered the fusion of frontal and profile face images, future research can explore the integration of additional modalities, such as thermal images or 3D facial scans. Combining these modalities can provide richer and more comprehensive information, potentially enhancing the system's ability to handle challenging scenarios with greater accuracy.
- **Exploration of Advanced Fusion Techniques:** This study employed feature-level and decision-level fusion techniques, which proved effective in improving recognition accuracy. However, there are other fusion strategies, such as score-level fusion or hybrid fusion methods, that could be explored to further boost the system's performance. Investigating novel fusion techniques and evaluating their impact on recognition accuracy would be a valuable direction for future research.
- **Robustness to Occlusions and Disguises:** In real-world scenarios, faces are often partially occluded by objects or individuals, and people may attempt to disguise their appearance. Future work can focus on developing techniques that are robust to occlusions and disguises, ensuring accurate recognition even in challenging conditions. This could involve incorporating attention mechanisms, adversarial

training, or advanced image inpainting algorithms to reconstruct occluded or disguised facial regions.

Appendix A

Poster

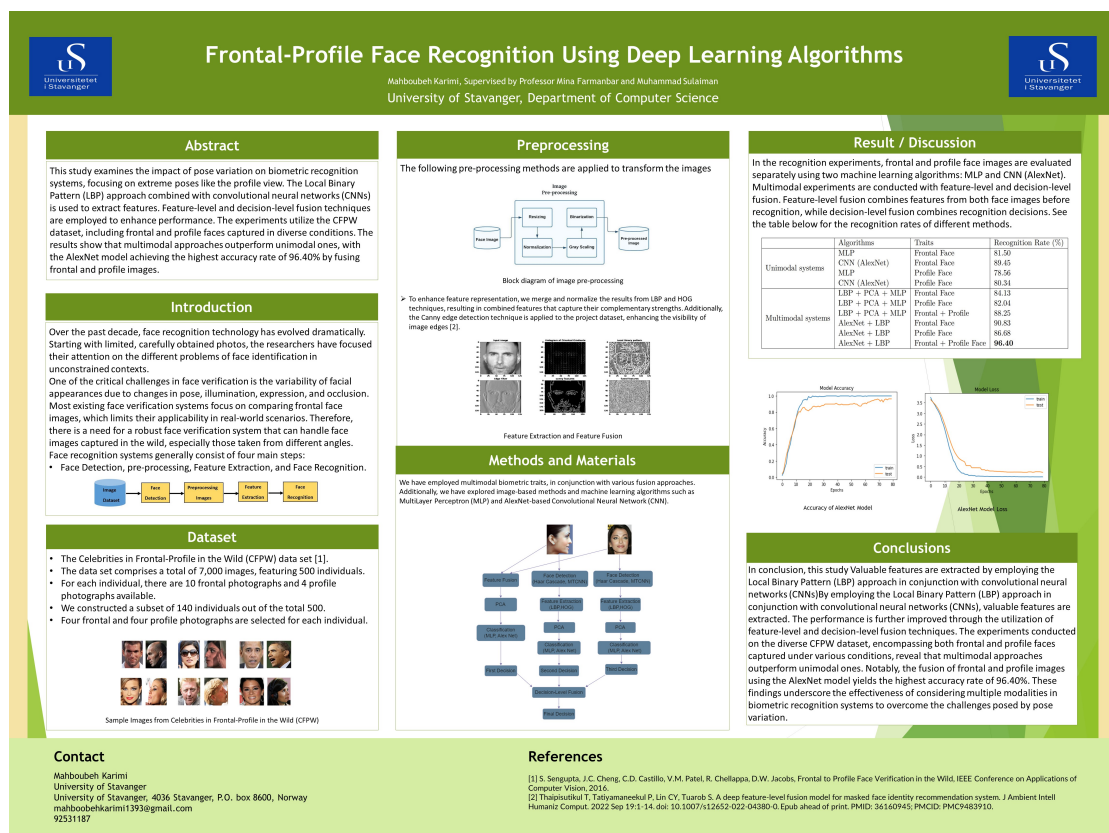


Figure A.1: The Poster

The code for the proposed method is available in the following GitHub Repository:
<https://github.com/mahboubeh1987/master-thesis-code>

Bibliography

- [1] Lei Yang, Jie Ma, Jian Lian, Yan Zhang, and Houquan Liu. Deep representation for partially occluded face verification. *EURASIP Journal on Image and Video Processing*, 2018, 12 2018. doi: 10.1186/s13640-018-0379-2.
- [2] Manisha Kasar, Debnath Bhattacharyya, and Tai-hoon Kim. Face recognition using neural network: A review. *International Journal of Security and Its Applications*, 10:81–100, 03 2016. doi: 10.14257/ijisia.2016.10.3.08.
- [3] C.D. Castillo V.M. Patel R. Chellappa D.W. Jacobs S. Sengupta, J.C. Cheng. Frontal to profile face verification in the wild. In *IEEE Conference on Applications of Computer Vision*, February 2016.
- [4] Utsav Prabhu, Jingu Heo, and Marios Savvides. Unconstrained pose-invariant face recognition using 3d generic elastic models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33:1952 – 1961, 11 2011. doi: 10.1109/TPAMI.2011.123.
- [5] Akshay Asthana, Tim K. Marks, Michael J. Jones, Kinh H. Tieu, and M. V. Rohith. Fully automatic pose-invariant face recognition via 3d pose normalization. *2011 International Conference on Computer Vision*, pages 937–944, 2011.
- [6] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003. doi: 10.1109/TPAMI.2003.1227983.
- [7] David Hoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16:2639–64, 01 2005. doi: 10.1162/0899766042321814.
- [8] Abhishek Sharma and David Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. volume 1, pages 593 – 600, 07 2011. doi: 10.1109/CVPR.2011.5995350.

- [9] Annan Li, Shiguang Shan, Xilin Chen, and Wen Gao. Maximizing intra-individual correlations for face recognition across pose differences. pages 605–611, 06 2009. doi: 10.1109/CVPR.2009.5206659.
- [10] Abhishek Sharma, Murad Al Haj, Jonghyun Choi, Larry S. Davis, and David W. Jacobs. Robust pose invariant face recognition using coupled latent space discriminant analysis. *Comput. Vis. Image Underst.*, 116:1095–1110, 2012.
- [11] Simon J.D. Prince, James H. Elder, Jonathan Warrell, and Fatima M. Felisberti. Tied factor analysis for face recognition across large pose differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):970–984, 2008. doi: 10.1109/TPAMI.2008.48.
- [12] P. Jonathon Phillips, Hyeonjoon Moon, Syed Rizvi, and Patrick Rauss. The feret evaluation methodology for face-recognition algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22:1090 – 1104, 10 2000. doi: 10.1109/34.879790.
- [13] Simon Prince, Peng Li, Yun Fu, Umar Mohammed, and James Elder. Probabilistic models for inference about identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):144–157, 2012. doi: 10.1109/TPAMI.2011.104.
- [14] Neeraj Kumar, Alexander Berg, Peter Belhumeur, and Shree Nayar. Attribute and simile classifiers for face verification. pages 365 – 372, 11 2009. doi: 10.1109/ICCV.2009.5459250.
- [15] Hieu Nguyen and Li Bai. Cosine similarity metric learning for face verification. volume 6493, pages 709–720, 11 2010. ISBN 978-3-642-19308-8. doi: 10.1007/978-3-642-19309-5_55.
- [16] Qiong Cao, Yiming Ying, and Peng Li. Similarity metric learning for face recognition. *2013 IEEE International Conference on Computer Vision*, pages 2408–2415, 2013.
- [17] Jiwen Lu, Junlin Hu, and Yap-Peng Tan. Discriminative deep metric learning for face and kinship verification. *IEEE Transactions on Image Processing*, 26(9): 4269–4282, 2017. doi: 10.1109/TIP.2017.2717505.
- [18] Gary B. Huang, Honglak Lee, and Erik G. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2518–2525, 2012.
- [19] Matthieu Guillaumin, Jakob J. Verbeek, and Cordelia Schmid. Is that you? metric learning approaches for face identification. *2009 IEEE 12th International Conference on Computer Vision*, pages 498–505, 2009.

- [20] Martin Köstinger, Martin Hirzer, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2288–2295, 2012.
- [21] Ge Wen, Huaguan Chen, Deng Cai, and Xiaofei He. Improving face recognition with domain adaptation. *Neurocomputing*, 287:45–51, 2018. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2018.01.079>. URL <https://www.sciencedirect.com/science/article/pii/S0925231218301127>.
- [22] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3025–3032, 2013.
- [23] Karen Simonyan, Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Fisher vector faces in the wild. In *British Machine Vision Conference*, 2013.
- [24] Haoxiang Li, Zhe Lin, Jonathan Brandt, and Jianchao Yang. Probabilistic elastic matching for pose variant face verification. pages 3499–3506, 06 2013. doi: 10.1109/CVPR.2013.449.
- [25] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.
- [26] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. 12 2014. doi: 10.1109/CVPR.2015.7298907.
- [27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.
- [28] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1:I–I, 2001.
- [29]
- [30] Cha Zhang and Zhengyou Zhang. A survey of recent advances in face detection. Technical Report MSR-TR-2010-66, June 2010.
- [31] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. 09 2014. ISBN 978-3-319-10598-7. doi: 10.1007/978-3-319-10599-4_7.

- [32] Alem Fitwi, Meng Yuan, Seyed Yahya Nikouei, and Yu Chen. Minor privacy protection by real-time children identification and face scrambling at the edge. *EAI Endorsed Trans. Security Safety*, 7:e3, 2020.
- [33] Tanoy Debnath, Md. Mahfuz Reza, Anichur Rahman, Amin Beheshti, Shahab S. Band, and Hamid Alinejad-Rokny. Four-layer convnet to facial emotion recognition with minimal epochs and the significance of data diversity. *Scientific Reports*, 12: 1–18, 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-11173-0.
- [34] Jesus Olivares-Mercado, Karina Toscano-Medina, Gabriel Sanchez-Perez, Mariko Nakano Miyatake, Hector Perez-Meana, and Luis Carlos Castro-Madrid. Face recognition based on texture descriptors. In Ricardo Lopez-Ruiz, editor, *From Natural to Artificial Intelligence*, chapter 6. IntechOpen, Rijeka, 2018. doi: 10.5772/intechopen.76722. URL <https://doi.org/10.5772/intechopen.76722>.
- [35] Tibor Trnovszký, Patrik Kamencay, Richard Orjesek, Miroslav Benco, and Peter Sykora. Animal recognition system based on convolutional neural network. *Advances in Electrical and Electronic Engineering*, 15:517–525, 2017.
- [36] Bilel Ameer, Sabeur Masmoudi, Amira Guidara Derbel, and Ahmed Ben Hamida. Fusing gabor and lbp feature sets for knn and src-based face recognition. *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pages 453–458, 2016.
- [37] Özgür Kaplan and Ediz Saykol. Comparison of support vector machines and deep learning for vehicle detection. 11 2018.
- [38] Abhishek Bansal, Kapil Mehta, and Sahil Arora. Face recognition using pca and lda algorithm. In *Proceedings of the 2012 Second International Conference on Advanced Computing & Communication Technologies, ACCT '12*, page 251–254, USA, 2012. IEEE Computer Society. ISBN 9780769546407. doi: 10.1109/ACCT.2012.52. URL <https://doi.org/10.1109/ACCT.2012.52>.
- [39] Shruti Sehgal, Harpreet Singh, Mohit Agarwal, V. Bhasker, and Shantanu. Data analysis using principal component analysis. In *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*, pages 45–48, 2014. doi: 10.1109/MedCom.2014.7005973.
- [40] Patrik Kamencay, Miroslav Benco, Tomas Mizdos, and Roman Radil. A new method for face recognition using convolutional neural network. *Advances in Electrical and Electronic Engineering*, 15:663–672, 2017.
- [41] Musab Coşkun, Ayşegül Uçar, Özal Yildirim, and Yakup Demir. Face recognition based on convolutional neural network. In *2017 International Conference on Modern*

-
- Electrical and Energy Systems (MEES)*, pages 376–379, 2017. doi: 10.1109/MEES.2017.8248937.
- [42] Dabiah Alboaneen, Hua Tianfield, and Yan Zhang. Glowworm swarm optimisation for training multi-layer perceptrons. pages 131–138, 12 2017. doi: 10.1145/3148055.3148075.
- [43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.
- [44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115, 09 2014. doi: 10.1007/s11263-015-0816-y.
- [45] Robert Mash, Nicholas Becherer, Brian Woolley, and John Pecarina. Toward aircraft recognition with convolutional neural networks. In *2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS)*, pages 225–232, 2016. doi: 10.1109/NAECON.2016.7856803.
- [46] Ashraf Mohra, Eman Zakaria, Wael Mohamed, and Abeer Khalil. Face recognition using deep neural network technique. 06 2019.