CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

A Comparative Evaluation Framework for Face Recognition from Surveillance Video Feeds

by

Nadim Zia

A thesis submitted in partial fulfillment for the degree of Master of Science

in the

Faculty of Electrical Engineering
Department of Electrical and Computer Engineering

2025

To my loving mother, in memory of my late father, to my dear sisters and brothers, and to little Noor — a joy and light in our lives.

CERTIFICATE OF APPROVAL

A Comparative Evaluation Framework for Face Recognition from Surveillance Video Feeds

by

Nadim Zia
(MEE233003)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Adil Masood Siddiqui	NUST, Islamabad
(b)	Internal Examiner	Dr. Nadeem Anjum	CUST, Islamabad

Dr. Imtiaz Ahmad Taj
Thesis Supervisor
November, 2025

Dr. Noor Muhammad Khan
Head
Dept. of Electrical Engineering
November, 2025

Dr. Imtiaz Ahmad Taj
Dean
Faculty of Engineering
November, 2025

Author's Declaration

I, **Nadim Zia** hereby state that my MS thesis titled "**A Comparative Evaluation Framework for Face Recognition from Surveillance Video Feeds**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

(Nadim Zia)

Registration No: MEE233003

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**A Comparative Evaluation Framework for Face Recognition from Surveillance Video Feeds**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Nadim Zia)

Registration No: MEE233003

Acknowledgement

First and foremost, I would like to dedicate this work to the loving memory of my late father, whose unwavering values, silent sacrifices, and lifelong encouragement continue to inspire me. His presence is deeply missed, but his teachings live on in everything I strive to achieve.

I am immensely grateful to my mother, whose constant love, prayers, and support have been a pillar of strength throughout this journey. Her belief in me has helped me persevere through every challenge.

I also extend my heartfelt thanks to all my family members for their patience, encouragement, and moral support during the entire research period.

I would like to express my sincere appreciation to all the members of the MBACS (Multi-Biometric Access Control System) project and the VisPRS research group. Their collaboration, technical input, and teamwork played a crucial role in the successful execution of this work.

A very special thanks to Dr. Imtiaz Ahmad Taj for his exceptional supervision and continuous guidance throughout the research period. His expertise, motivation, and insightful feedback have been instrumental in shaping the direction and quality of this research.

This work would not have been possible without the valuable support and contributions of all the individuals and teams mentioned above.

(Nadim Zia)

Abstract

The global facial recognition market is projected to reach $16.74 billion by 2030, driven by the growing need for secure and intelligent biometric authentication systems. However, achieving accurate face recognition in surveillance videos, often involving motion blur, low resolution, and uncontrolled lighting, remains a significant challenge. In this study, we present a surveillance video-based face recognition pipeline and conduct a detailed comparative performance evaluation of 16 model combinations. These combinations result from pairing four state-of-the-art face recognition models: ArcFace with a ResNet backbone, ArcFace with a MobileNet backbone, AdaFace, and a Sub-center Contrastive Distillation model, with four face detection models: MTCNN and three variants of RetinaFace. Evaluations were performed on both benchmark datasets and a custom surveillance dataset. Among all combinations, the pairing of AdaFace with RetinaFace (ResNet backbone) consistently achieves the best performance in terms of recognition accuracy and robustness under real-world conditions. Furthermore, we extend the study by evaluating the performance of Convolutional Neural Network (CNN)-based models and Vision Transformer (ViT)-based models for face recognition. While ViT-based models demonstrate slightly better accuracy on certain datasets, the performance gain is often negligible. When additional parameters such as model complexity, computational overhead, and memory requirements are taken into account, CNN-based models remain competitive and, in many scenarios, more practical for real-time deployment. Our comprehensive comparative analysis highlights that ViTs do not definitively surpass CNNs in surveillance-based face recognition when a broader set of evaluation metrics is considered. This study provides practical insights into selecting effective and balanced model architectures for real-world face recognition systems.

Contents

List of Figures

List of Tables

Abbreviations

AdaFace	Adaptive Margin Face Loss
ArcFace	Additive Angular Margin Loss
AUC	Area Under the Curve
Bboxes	Bounding Boxes
CNN	Convolutional Neural Network
EER	Equal Error Rate
FPR	False Positive Rate
FR	Face Recognition
FPS	Frame Per Second
GPU	Graphics Processing Unit
KD	Knowledge Distillation
LFW	Labeled Faces in the Wild
MBACS	Multi Biometric Access and Control System
MTCNN	Multi-task Cascaded Convolutional Neural Network
MS1M	MS-Celeb-1M Dataset
NMS	Non Maximum Suppression
ResNet	Residual Network
ROC	Receiver Operating Characteristic
RoI	Region of Interest
TPR	True Positive Rate
ViT	Vision Transformer
YOLO	You Only Look Once (Object Detector)

Chapter 1

Introduction

1.1 Face Recognition

Biometrics involves the automated identification of human physiological and behavioral characteristics. In recent decades, advancements in biometric technology have led to the widespread industrial adoption of systems based on various modalities, including facial recognition, iris scanning, gait analysis, fingerprinting, and palmprint identification. Among these, facial recognition is particularly popular due to its contactless acquisition, noninvasive nature, social acceptance, and effectiveness in non-cooperative situations. These innovations have facilitated their deployment across diverse domains such as access control, surveillance, and identity verification. Nonetheless, ongoing research continues to address critical concerns related to data privacy, algorithmic fairness, and susceptibility to adversarial attacks.

1.2 Face Recognition Pipeline

The development of face recognition models using deep learning algorithms consists of two main phases: training and inference. During the training phase, preprocessing and model training occur, allowing the network to learn to extract

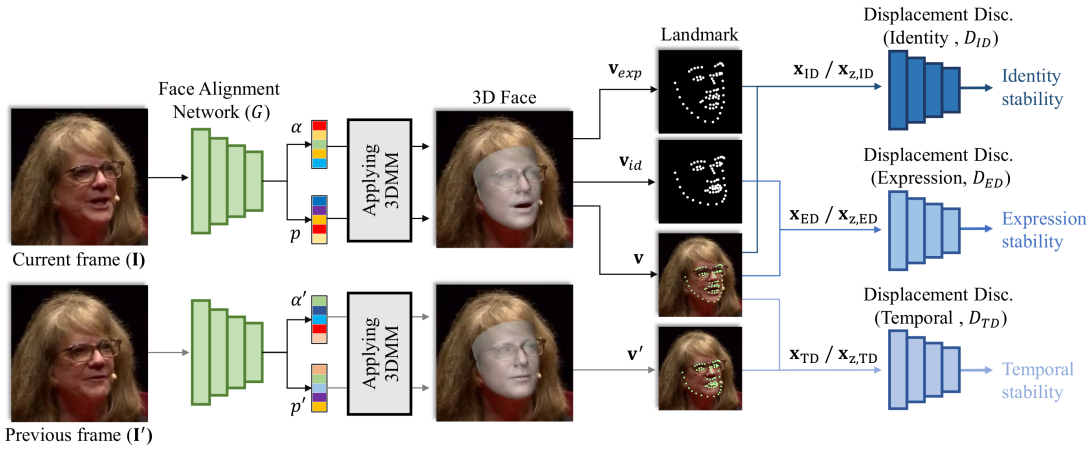


Figure 1.1: A Robust Face Alignment Approach Presenting the Basic Idea behind Alignment[1].

distinctive features from facial images. In contrast, the inference phase involves preprocessing, feature extraction, and matching, using the trained model to recognize faces in new images.

1.2.1 Preprocessing

Before an image is processed by a face recognition model, it must undergo a crucial preprocessing step that includes face detection and alignment. Face detection is carried out using Face Localization Algorithms like MTCNN (Multi-task Cascaded Convolutional Neural Networks) and RetinaFace, which typically provide the coordinates of a bounding box along with the landmarks associated with the face. lignment ensures that the face is consistently positioned by rotating the image so that both eyes are on a horizontal line. Figure 1.1 illustrates an approach to face alignment, highlighting both the process of alignment and its significance in ensuring accurate facial analysis.

1.2.1.1 Benefits of Alignment

- Improved Comparability By aligning faces to a standardized pose and scale, facial features become more comparable across different faces. The alignment part ensures that the comparision being done is just.

- Robustness to Variations Alignment helps reduce the effects of variations in pose, expression, and lighting, making the features more consistent and easier to analyze. This ensures that facial embeddings extracted from different conditions remain comparable across subjects. Moreover, it enhances recognition accuracy by minimizing intra-class variation. Proper alignment also stabilizes feature extraction across datasets with diverse imaging conditions.

1.2.2 Feature Extraction

Feature extraction is crucial for the effectiveness of face recognition systems. In the training phase, the face recognition model, typically a CNN, is trained to learn distinctive features. This process involves using an additional classification layer and a loss function to train the model. In the testing phase, the classification layer is removed, and the extracted features are used for matching.

1.2.3 Loss Function

Let's proceed step by step to understand how ArcFace has formulated a state-of-the-art loss function that simultaneously increases inter-class distance and reduces intra-class variance. The proposed approach of ArcFace is based on the well-known and widely used softmax loss function. Like other loss functions, ArcFace updates various parameters and performs a normalization step by step to transform the features into a new feature space, which is the surface of a hypersphere in k-dimensional space. This transformation allows for better separation and discrimination of features in the face recognition task. The softmax function is given by the equation:

$$P(y = i|\mathbf{x}) = \frac{e^{\mathbf{w}_i^T \mathbf{x} + b_i}}{\sum_{j=1}^{C} e^{\mathbf{w}_j^T \mathbf{x} + b_j}} \tag{1.1}$$

- $\mathbf{x}$ is the input feature vector extracted automatically by the feature extractor (e.g., the embeddings of a face image).

- $\mathbf{w}_i$ is the weight vector associated with class i. Every feature has a corresponding weight which is then multiplied by its weight.
- b_i is the bias term associated with class i.
- $P(y = i|\mathbf{x})$ is the probability that the input $\mathbf{x}$ belongs to class i.
- C is the number of classes.
- b_i is the bias term associated with class i.

1.2.3.1 Why Softmax for Probability Calculation?

Softmax is used for calculating probabilities of logits because it converts raw scores (logits) into a probability distribution, where the probability of each class is between 0 and 1, and the sum of all probabilities equals 1. This makes the output interpretable as a valid probability distribution over the possible classes. In contrast, a simpler function like sigmoid is used in binary classification and cannot provide a valid distribution across multiple classes. Sigmoid would independently map each logit to a probability between 0 and 1, without taking into account the relationships between the logits, which is not suitable for multi-class classification.

Hence, softmax is preferred because it ensures that the probabilities are normalized and mutually exclusive, making it ideal for problems like face recognition with multiple possible classes. Furthermore, softmax facilitates a clear decision boundary by emphasizing the most confident prediction among competing classes. This property improves classification and also its generalizability, especially in deep networks where inter-class similarity can otherwise lead to ambiguous outputs.

1.2.3.2 Feature Transformations

In this part, the features are transformed from a low-dimensional space to a higher-dimensional space. This is achieved by projecting feature vectors onto the surface of a hypersphere by normalizing both the feature vector and the weight matrix. As Iknow, cosine similarity can be expressed as the dot product between the feature

matrix and the weight matrix when both vectors have magnitudes equal to 1. This normalization eliminates the effect of varying magnitudes of the feature and weight matrices on the embeddings. Now, the embeddings are purely dependent on the angle between the feature vector and its corresponding true class. Mathematically, cosine similarity is defined as:

$$\text{cosine similarity} = \frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|\|\mathbf{x}\|} \tag{1.2}$$

When both the feature vector $\mathbf{x}$ and the weight vector $\mathbf{w}$ are normalized to unit length (i.e., $\|\mathbf{w}\| = 1$ and $\|\mathbf{x}\| = 1$), the cosine similarity between the feature and weight vector simplifies to:

$$\text{cosine similarity} = \mathbf{w}^T\mathbf{x} = \cos(\theta) \tag{1.3}$$

where θ is the angle between the characteristic vector $\mathbf{x}$ and its corresponding true class weight vector, which is usually one of the columns of the matrix $\mathbf{W}$. Now, the embeddings lie on the surface of a hypersphere, and their values depend solely on the angle between the feature vector and its true class (ground truth). The size of the hypersphere is determined by the number of dimensions, k. As the number of dimensions, k, increases, the surface area of the sphere grows. For example, in a 3-dimensional space, the surface area of the sphere is 4π, while for a 4-dimensional sphere, the surface area is $2\pi^2$. ArcFace uses 512 dimensions by default, but this number can vary depending on the number of identities. There should be a trade-off between the number of identities and the number of dimensions of the hypersphere. If the number of identities increases without adjusting the dimensionality of the sphere, the surface area of the hypersphere will saturate, and there will be insufficient space for new embeddings.

1.2.3.3 Normalization and Scaling

After the feature normalization and projection to the hypersphere, the dot product $\mathbf{w}^T\mathbf{x}$ simplifies due to normalization of both the feature vector $\mathbf{x}$ and the weight matrix $\mathbf{W}$ to have unit magnitudes. In this way, the effect of magnitude on the

embeddings has been eliminated; hence, only the angle between the individual weight and the feature determines the discriminative power of the representation. Such geometric interpretation of features aligns with the concept of margin-based learning, promoting tighter intra-class similarity and larger inter-class margins. Consequently, the learned embedding space becomes more structured and interpretable, benefiting downstream face verification and identification tasks. This angular-based separation ensures more robust class boundaries, improving the model's generalization ability across diverse face samples. The model hence obtained will be much more robust against every type of variation. It is equal to the cosine of the angle in:

$$\cos(\theta) = \frac{\mathbf{w}^T\mathbf{x}}{|\mathbf{w}||\mathbf{x}|} \tag{1.4}$$

This cosine similarity is then scaled by a term s to improve the separation of features in the hypersphere:

$$s \cdot \cos(\theta) \tag{1.5}$$

To further improve the feature discriminability, ArcFace removes the bias terms as the hypersphere is not biased. The final normalized softmax loss (after scaling and normalization) can be written as:

$$L_{\text{softmax\_norm}} = -\log\left(\frac{\exp(s\cos(\theta_y))}{\sum_{i=1}^{N}\exp(s\cos(\theta_i))}\right) \tag{1.6}$$

1.2.3.4 Additive Angular Margin

The addition of the angular additive margin is the core innovation introduced by ArcFace. After normalization and scaling, the feature vectors now lie on the surface of the hypersphere. The core idea is to add an angular margin to the angle between the feature vector and the corresponding ground-truth class. This addition of margin provides two main benefits: it increases the inter-class distance and decreases the intra-class distance, resulting in high-quality embedding generation. It can be written as:

$$L_{\text{ArcFace}} = -\log\left(\frac{\exp(s\cos(\theta_y + m))}{\exp(s\cos(\theta_y + m)) + \sum_{i\neq y}\exp(s\cos(\theta_i))}\right) \tag{1.7}$$

Where:

- $\cos(\theta_y)$ is the cosine similarity between the feature vector and the true class center, after normalization and scaling.
- $\cos(\theta_i)$ is the cosine similarity between the feature vector and the other class centers, after normalization and scaling.
- s is the scaling factor, which controls the magnitude of the cosine similarity to enhance feature discriminability.
- m is the angular margin added to the true class angle, which increases the inter-class distance and reduces the intra-class distance, enhancing feature separability.

This loss function enforces a larger angular margin between the true class and other classes, ensuring that the embeddings lie on the surface of the hypersphere and are influenced primarily by the angle between the feature vector and its true class. Embeddings generated by ArcFace exhibit significantly higher inter-class separability, while the intra-class distance is notably reduced. This results in more discriminative embeddings, which are crucial for accurate face recognition.

1.2.4 Matching

During the matching stage, the features extracted from probe images are compared with those from the gallery set images. This comparison is typically performed using cosine similarity or Euclidean distance. A smaller distance or higher cosine similarity indicates a closer match between the probe and gallery embeddings, suggesting that both belong to the same identity. To further improve reliability, normalization of embeddings is often applied before distance computation to maintain consistent vector magnitudes. Additionally, threshold tuning is crucial to balance false acceptance and rejection rates across varying environmental conditions.

1.3 Backbone Architectures for Face Recognition and Detection

1.3.1 ResNet

ResNet, or Residual Network, is a convolutional neural network (CNN) architecture developed to overcome the difficulties associated with training very deep networks. In traditional CNNs, vanishing gradients can hinder the learning process in deeper layers, preventing the network from effectively capturing complex features. ResNet solves this issue by using skip connections, which bypass a set of layers and add the input directly to the output. This enables the network to learn identity mappings alongside the transformations from the layers, facilitating gradient flow and supporting the training of extremely deep architectures. The core component of ResNet is the residual block, which includes convolutional layers, batch normalization, and activation functions, with a skip connection linking the input to the output of the block. ResNet has multiple variants such as ResNet-50 and ResNet-101, which differ in depth by adjusting the number of layers within each residual block. This architecture allows for significantly deeper networks that capture more intricate patterns in data. ResNet offers several advantages, including the ability to train deeper networks, improved accuracy on tasks such as image classification and object detection, and broad applicability across different domains by modifying the final layers.

1.3.2 MobileFaceNet

MobileFaceNet is a convolutional neural network (CNN) architecture specifically developed for facial recognition tasks on mobile devices. Unlike traditional face recognition models that often require significant computational resources, MobileFaceNet is designed to be lightweight and efficient while maintaining a high level of accuracy. This makes it particularly suitable for applications such as mobile security and facial unlocking on smartphones, where performance and speed are

critical. The architecture incorporates depthwise separable convolutions to reduce the number of parameters and computational cost compared to standard CNNs. It also uses a bottleneck structure with residual connections to extract high-quality facial features efficiently.

A distinctive feature of MobileFaceNet is the use of global depthwise convolution, which replaces traditional average pooling layers to capture global feature information with fewer parameters. The model typically processes fixed-size input images, such as 112x112 pixels, and produces a compact 512-dimensional feature vector that represents the face, enabling efficient storage and fast comparison for recognition purposes. MobileFaceNet offers real-time performance on mobile platforms due to its streamlined architecture. It strikes an effective balance between accuracy and computational efficiency, making it a practical solution for deployment on resource-constrained devices. Its compact nature and speed enable robust facial recognition capabilities in mobile environments, making it a key choice for enhancing mobile security and user authentication systems.

1.4 Research Motivation

The demand for reliable and secure face recognition systems has significantly increased due to the rapid rise in surveillance applications, biometric authentication, and public security concerns. In high-risk environments such as airports, public transit systems, and smart cities, facial recognition plays a crucial role in identity verification and threat detection. However, real-world surveillance footage typically presents challenges such as low resolution, motion blur, varying lighting conditions, and occlusions, making robust face recognition a complex task.

Traditionally, Convolutional Neural Networks (CNNs) have demonstrated impressive performance in face recognition tasks. Architectures like ArcFace, AdaFace, and MobileFaceNet have become widely adopted due to their high accuracy and efficiency. More recently, Vision Transformers (ViTs) have emerged as a strong alternative, offering the ability to model long-range dependencies and capture richer

global features. Preliminary studies indicate that ViTs can outperform CNNs in some cases, but questions remain regarding their computational complexity, model size, and generalizability under surveillance conditions.

This research is motivated by the need to conduct a fair and rigorous comparison between state-of-the-art CNN-based and ViT-based face recognition models. While ViTs may offer slightly higher accuracy, it is essential to consider other practical parameters such as inference time, memory consumption, and deployment feasibility in resource-constrained environments. Moreover, there is a lack of extensive comparative evaluations that incorporate both benchmark datasets and real-world surveillance footage. This thesis aims to fill that gap by designing and implementing a comprehensive framework that evaluates multiple face detection and recognition model combinations. The findings will provide valuable insights for researchers and practitioners seeking to deploy efficient and accurate face recognition systems in surveillance-based scenarios.

1.5 Research Objectives

The primary aim of this research is to conduct a comprehensive comparative analysis of face recognition models—spanning both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs)—under real-world surveillance conditions. In order to systematically address the mentioned aim stated, the following objectives have been defined:

1. To design and develop a comprehensive face recognition framework
   The first objective is to architect a modular and scalable face recognition pipeline tailored for surveillance scenarios. The framework integrates face detection, alignment, and recognition stages, allowing flexibility to plug and evaluate various state-of-the-art models.

2. To implement and analyze multiple state-of-the-art face recognition algorithms

The second objective focuses on the selection, implementation, and fine-tuning of leading face recognition models. This includes CNN-based models such as ArcFace (with ResNet and MobileNet backbones), AdaFace, and Sub-center Contrastive Distillation networks, as well as Transformer-based approaches. The goal is to understand their performance in terms of accuracy, computational efficiency, and suitability for constrained environments such as real-time surveillance.

3. To assess the robustness and reliability of the models under varying conditions

   Real-world surveillance footage introduces challenges such as motion blur, occlusion, illumination variance, and low resolution. This objective involves evaluating how well each model performs under these uncontrolled and often adverse conditions.

By addressing these objectives, the study aims to deliver an insightful and actionable understanding of the trade-offs between CNN and ViT-based approaches for face recognition in practical deployment scenarios. The ultimate goal is to guide future design choices for robust, accurate, and efficient face recognition systems tailored to the demands of real-world surveillance.

1.6 Research Goals

The goals of this research are outlined below to provide a clear direction for the study:

- To explore and understand the evolving landscape of face recognition technologies, particularly comparing Convolutional Neural Networks (CNNs) with Vision Transformers (ViTs).
- To contribute to the development of more accurate and efficient face recognition systems for surveillance applications through in-depth analysis and benchmarking, which can be used for future work.

- To identify the strengths, limitations, and trade-offs between CNN-based and ViT-based models, not only in terms of accuracy but also in terms of computational complexity, inference time, model size, and real-world robustness.
- To provide a practical guide for researchers and practitioners in selecting the most suitable face recognition models based on application-specific constraints and requirements.
- To enhance the reliability and trustworthiness of face recognition systems operating in uncontrolled environments such as low-resolution or motion-blurred surveillance videos.

1.7 Thesis Structure

This thesis is organized into the following chapters to ensure a systematic and coherent presentation of the research work:

- Chapter 1 – Introduction This chapter provides an overview of the research problem, its significance, and the motivation behind conducting a comparative analysis of face recognition models, including state-of-the-art models from both CNN and vision transformer domains. It also outlines the research objectives and goals.
- Chapter 2 – Literature Review A comprehensive review of existing face recognition technologies is presented, including face detection models such as MTCNN and RetinaFace, and face recognition models from both CNN and ViT domains. The chapter also identifies research gaps and formulates the problem statement that guides the proposed study. The literature review provides the preceding concept of the same research.
- Chapter 3 – Proposed Technique and Methodology This chapter describes the complete face recognition pipeline developed in this research. It includes

preprocessing techniques, optimization steps, model selection, and integration strategies. Details about the framework design and system flow are also discussed.

- Chapter 4 – Implementation and Results All experimental setups and evaluation protocols are described in this chapter. Multiple combinations of face detectors and recognition models are tested on benchmark and custom surveillance datasets. The performance metrics are presented using graphs and tables for better visualization and comparison.

- Chapter 5 – Conclusion and Recommendations This chapter summarizes the key findings of the research. It highlights the comparative performance outcomes and discusses practical implications. Recommendations for future research directions are also provided based on observed limitations and insights.

- References A complete list of bibliographic references used throughout the thesis is provided, following standard citation formats. Corresponding URLs have also been provided where applicable.

1.8 Datasets

Datasets play a key role in the performance of face recognition models. For convenience, they are divided into training and testing datasets. Training datasets are used to train the model, while testing datasets are used for evaluation purposes. For evaluation, standard benchmark datasets have been utilized to measure the model's performance against established baselines.

1.8.1 Training Datasets

- CASIA Webface Casia WebFace is automatically collected from the web by the CASIA group and then manually refined. It contains 0.5 million images of 10,000 celebrities. The famous subjects have more images, while a few

describe other subjects. CASIA Web Face is considered a small-scale dataset in face recognition.

- MS Celeb 1M The Microsoft Celeb 1M (MS Celeb 1M) dataset is a large-scale collection of facial images designed to advance face recognition research. It contained 10 million images of 1 million celebrities collected from the internet.

- VGGFace2 The VGGFace2 dataset comprises over 3.3 million images of 9,000+ individuals, captured under diverse conditions of pose, illumination, and age. It is widely used for training deep learning models due to its high intra-class variation and strong generalization capability.

1.8.2 Testing Dataset

For the development and evaluation System , a comprehensive testing dataset was collected from 48 students. The dataset includes facial images, hand images, and personal information. Video footage was captured using four strategically positioned cameras to ensure comprehensive coverage from different angles.

1.8.2.1 Camera Positions

- Top of Pool Camera (Camera 1):Positioned at the top of a rod pool at a height of 7 feet, with an additional 20 inches (1.67 feet) added.

- Side Cameras (Camera 2, Camera 3, Camera 4): Positioned on the rod pool at a height of 7 feet. The cameras are placed 6 feet apart from each other.

- Distance Markers: 5 ft, 10 ft, 15 ft.

1.8.2.2 Marked Paths

Specific paths were marked on the ground to guide the movement of the students during video capture: Path 1: The student walks straight toward the camera.

Path 2: The student moves from the left side toward the camera. Path 3: The student moves from the right side toward the camera. Distance Marking Points: Points were marked on the ground at: 5 feet 10 feet 15 feet These markings were used to calculate the distance of the students from the cameras during the video capture process.

1.8.3 Evaluation Datasets

This table This table 1.1 provides an extensive overview of key benchmark datasets commonly used to evaluate face recognition models under diverse real-world conditions, such as variations in age, pose, illumination, and image quality. The AgeDB-30 dataset consists of 12,240 facial images of 440 subjects, with an average 30-year age gap between image pairs, making it ideal for assessing age-invariant recognition capabilities. The LFW (Labeled Faces in the Wild) dataset includes over 13,000 images of 5,749 individuals collected from the web and serves as the most widely adopted benchmark for unconstrained face verification. The CPLFW dataset extends LFW by introducing greater pose variations, thereby testing the robustness of models to head orientation changes. Meanwhile, the CPF-FF and CPF-FP subsets, derived from the CPF dataset, specifically evaluate front-front and front-profile matching conditions, which are crucial in surveillance and non-frontal scenarios. CALFW further enhances LFW by incorporating noticeable age gaps between image pairs, providing a strong benchmark for evaluating cross-age recognition. The VGGFace2 dataset, containing over 3.3 million images across 9,000 identities, captures wide variations in pose, illumination, expression, and ethnicity, making it highly suitable for both training and large-scale evaluation. Collectively, these datasets create a diverse and balanced suite for comprehensive benchmarking of face recognition models across multiple challenges.

In addition to scale and diversity, these datasets differ significantly in annotation accuracy, image resolution, and demographic balance, which directly influence model training outcomes. The inclusion of both controlled and in-the-wild datasets allows researchers to assess how models generalize across different data

domains. Standardized evaluation protocols, including 5-fold and 10-fold cross-validation, ensure fair and reproducible performance assessment. Recent works have also proposed the use of synthetic and semi-synthetic datasets to fill gaps in demographic representation and mitigate bias. Combining multiple datasets provides a holistic understanding of recognition performance across variations in pose, lighting, and occlusion.

Furthermore, emerging benchmark suites are beginning to include temporal and video-based datasets, which evaluate recognition stability under motion and frame continuity. Some studies also integrate multimodal modalities such as depth, thermal, or infrared imaging to improve recognition under low-light and occluded conditions.

Table 1.1: Comparative Summary of Benchmark Face Verification Datasets

Dataset	**Full Name**	**Description**	**Purpose**
AgeDB-30	Age Database (30-year split)	Contains face pairs with large age gaps (∼30 years apart).	Age-invariant verification
LFW	Labeled Faces in the Wild	13,000+ images collected from the web under unconstrained conditions.	Unconstrained verification
CPLFW	Cross-Pose Labeled Faces in the Wild	Extension of LFW emphasizing larger pose variations.	Pose-robust verification
CPF-FF	Cross-Pose Face – Front-Front	Pairs where both images are front-facing, sampled from CPF dataset.	Pose-consistent verification (easy)
CPF-FP	Cross-Pose Face – Front-Profile	Pairs where one image is front-facing and the other is profile view.	Pose-inconsistent verification (hard)
CALFW	Cross-Age Labeled Faces in the Wild	Extension of LFW with positive pairs having large age gaps.	Age-robust verification
VGG2-FP	VGGFace2 Front-Profile	Subset of VGGFace2 emphasizing front vs. profile identity verification.	Pose-invariant verification

Chapter 2

Literature Review

2.1 Introduction

Face recognition has become a cornerstone of biometric authentication and surveillance systems. Over the past two decades, there has been a paradigm shift from classical methods like Eigenfaces and Local Binary Pattern Histograms (LBPH) to deep learning-based approaches, notably Convolutional Neural Networks (CNNs). Recently, Vision Transformers (ViTs) have emerged as powerful alternatives due to their ability to model global dependencies. However, existing literature reveals a lack of comprehensive, head-to-head comparisons between these paradigms, especially in real-world surveillance scenarios. This literature review surveys the key contributions in face recognition research, identifies prevailing methodologies, and highlights a significant gap in comparative evaluation frameworks.

The face recognition pipeline consists of Face Detection, followed by Face Recognition. For face detection in videos, well-known algorithms such as the Adaboost-based Viola and Jones algorithm and the more recent Multi-Task Cascaded Convolutional Neural Networks (MTCNN) algorithm [2], can be evaluated to determine the optimal technique for this system. Face recognition, a highly researched topic in computer vision, has garnered significant attention due to its non-invasive nature. Controlled environment face recognition, involving frontal face images with

consistent backgrounds, has matured significantly, yielding high accuracy in applications such as registration. Early face recognition research focused on subspace representations such as Eigenfaces, Fisherfaces, Independent Component Analysis (ICA), and Laplacianfaces. [3–5] These algorithms, however, achieved only around 80% accuracy on frontal faces. This accuracy in applications related to the Face Recognition task, like an attendance management system, cannot be accepted. Subsequent generations of face recognition algorithms leveraged geometrical features such as Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG), [6–8], improving accuracy to over 90% for frontal images, but their performance deteriorated with variations in pose and expression. These methods relied heavily on handcrafted feature extraction, making them sensitive to illumination changes and occlusions. Furthermore, their limited ability to capture high-level abstract representations restricted their applicability in unconstrained environments. This motivated the shift toward learning-based and deep feature extraction techniques that could adapt to complex visual variations more effectively. The advent of deep learning architectures marked a significant leap in face recognition performance. The first major contribution was Facebook's Deep Face [9], a deep convolutional neural network model achieving 97.35% accuracy on frontal images. Google AI's Facenet [10]further advanced the field by introducing a triplet loss function to learn discriminant features, using a vast dataset of 200 million face images to achieve state-of-the- art performance. Subsequent models like L-SoftMax , A-SoftMax (Sphereface), Norm Face [11], Cos face [12], and Arc face introduced various modifications to improve feature discrimination and accuracy, with results ranging from 96% to 99% on standard face datasets. Convolutional neural networks play a strong role in face recognition applications. Despite these advancements, "face in the wild" scenarios, involving images captured from surveillance or CCTV cameras, present challenges such as low quality, blurriness, and varying expressions and poses. Accuracy for these conditions remains below 80%. Face normalization, a preprocessing step to improve matching accuracy, has evolved from simple alignment techniques to advanced methods involving Generative Adversarial Networks (GANs) and Autoencoders [13, 14] reconstructing high-quality facial representations. Recent approaches using Knowledge

Distillation and Teacher-Student architectures have shown improvements in accuracy for uncontrolled environments [14**?** –16] though these algorithms are often database-specific. Moreover, the effectiveness of these techniques heavily depends on illumination consistency and the availability of diverse training samples. Thus, designing effective face recognition models that maintain robustness across unseen domains remains an active and challenging area of research.

2.2 Classical Face Recognition Approaches

Earlier face recognition systems relied heavily on handcrafted feature extractors. Techniques like Eigenfaces and Fisherfaces used Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to extract relevant facial features. Local Binary Pattern Histograms (LBPH) and Gabor filters further improved robustness to illumination changes. A study titled *Comparative Analysis of AI Facial Recognition Algorithms* [17] contrasted these traditional techniques with early CNN-based models. While these methods were lightweight and interpretable, they failed to generalize to unconstrained environments, thus making way for deep learning-based approaches.

2.3 CNN-Based Face Recognition Models

CNNs revolutionized face recognition by learning hierarchical feature representations from raw pixels. Popular architectures such as FaceNet, VGGFace, and ArcFace dominated benchmarks due to their superior performance. These models typically employed triplet loss or softmax variants for embedding learning. A study [18] analyzing surveillance scenarios employed only CNN-based architectures with Softmax-based classification however it does not incorporates other SOTA models form various domain like ViTs. This limitation restricts the comprehensive evaluation of hybrid or transformer-based approaches that could offer improved feature representation and robustness. Therefore, integrating ViTs alongside CNNs

in future research could provide deeper insights into performance trade-offs under real-world surveillance conditions. Such comparative studies are essential to establish a unified understanding of how different architectures perform across varying datasets and operational constraints.

Another paper [19] reviewed CNN pipelines using custom datasets, noting their effectiveness in real-time surveillance but highlighting limitations such as susceptibility to adversarial noise and high dependency on training data volume. Moreover, the framework InsightFace remains a leading implementation of CNN-based recognition, although it shows signs of aging compared to modern frameworks. Recent studies have emphasized the need to integrate lightweight architectures and robust normalization layers to enhance CNN resilience under varying conditions.

2.4 Face Detectors

2.4.1 MTCNN

After extracting frames from the video, the next critical step is to locate the faces in each frame and extract these regions, often referred to as Region of Interest (ROI) detection [20]. This process is effectively carried out using MTCNN, or Multi-task Cascaded Convolutional Networks, which is a highly efficient face detection algorithm known for its lightweight CNN architecture, as shown in figure 2.1, making it suitable for real-time applications. The MTCNN algorithm operates through the following three stages, as shown in Figure 2.2:

1. The P-Net is the initial stage of MTCNN [21] and is responsible for generating candidate face regions. It takes an image as input and applies a series of convolutional layers to extract features. The output of the P-Net [21] consists of two main components: face/non-face classification scores and bounding box regression values. The face classification scores indicate the likelihood of a face being present at each position in the image. The bounding box regression values adjust the initially proposed bounding boxes to

better fit the detected faces. These outputs are used to generate a set of candidate face regions along with their associated confidence scores.

2. The R-Net takes the candidate face regions generated by the P-Net and refines them to improve accuracy. It further filters out false positives and adjusts the bounding boxes to better fit the detected faces. Similar to the P-Net, the R-Net [23] outputs face classification scores and refined bounding box coordinates. The refinement process helps to reduce the number of false positives and improve the accuracy of face detection.

3. Output Network (The O-Net is the final stage of MTCNN [24] and performs the most accurate face detection. It takes the refined candidate face regions from the R-Net and further refines them. In addition to face classification scores and bounding box coordinates, the O-Net [25] also predicts facial landmarks such as the positions of eyes, nose, and mouth corners. These facial landmarks are used for face alignment, which is essential for tasks such as face recognition. The final output of the O-Net consists of detected face regions along with their associated bounding boxes and facial landmarks.

Overall, the ROI separation process in PyTorch [26] MTCNN involves a series of stages where candidate face regions are progressively generated, refined, and filtered to accurately detect and align faces in an image.

2.4.2 RetinaFace

RetinaFace is a robust and efficient face detection algorithm developed to deliver high accuracy and real-time performance. Built on a deep learning framework, it is capable of detecting faces and their key landmarks with remarkable precision. RetinaFace is implemented using different backbone networks such as ResNet, Slim, and MobileNet, each offering a distinct trade-off between accuracy and computational efficiency depending on the deployment requirement. One of the core strengths of RetinaFace lies in its single-stage detection approach. Unlike multi-stage detectors, it simultaneously predicts bounding boxes and facial landmarks in

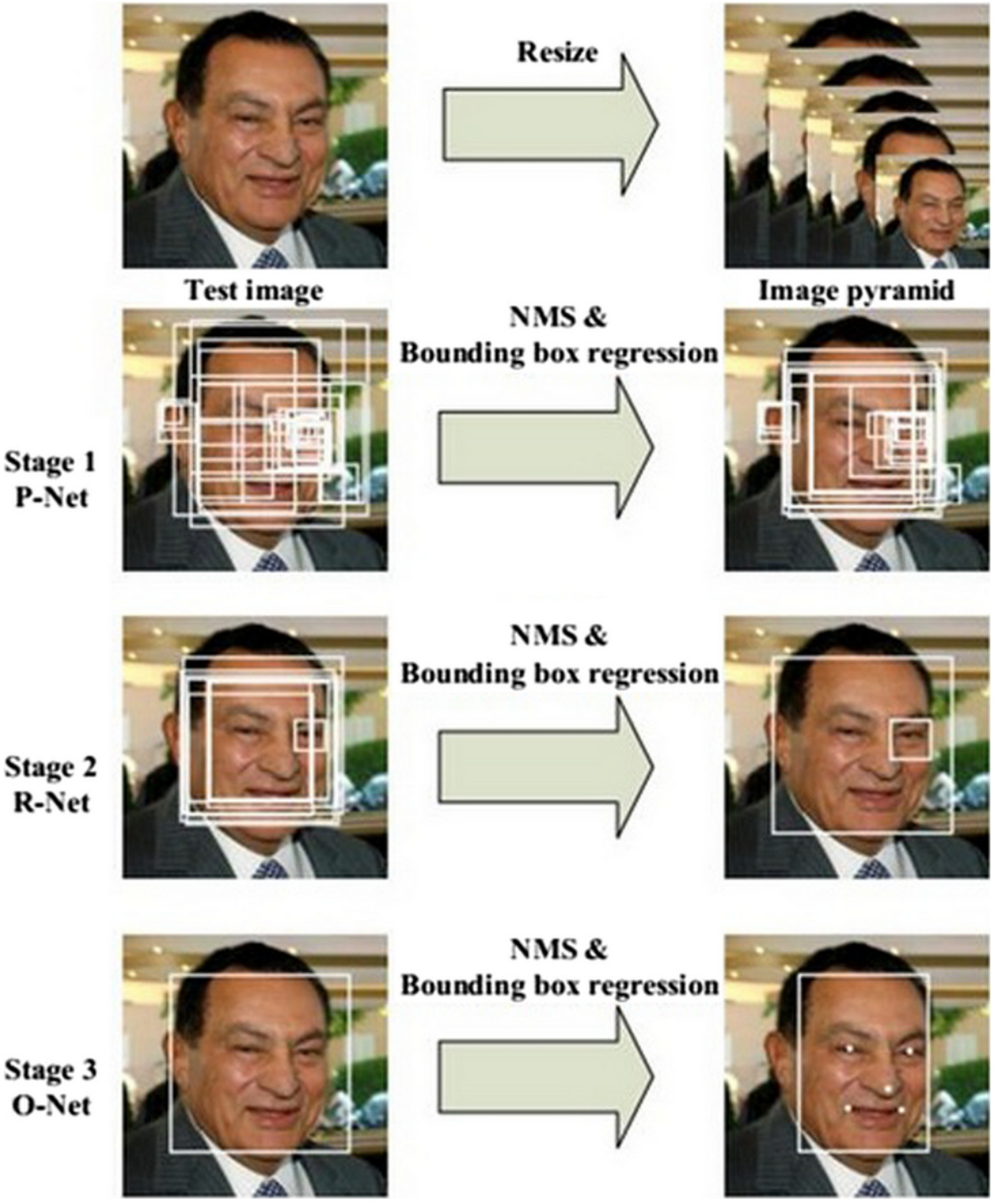


Figure 2.1: Visualization of Stage 1–3 in MTCNN. The first stage (P-Net) generates candidate face regions using a fully convolutional network. The second stage (R-Net) refines these proposals by rejecting false positives and improving bounding box accuracy. The third stage (O-Net) further fine-tunes detections and predicts facial landmarks for alignment. Together, these stages enable accurate multi-scale face detection across varying poses and lighting conditions [22]. Additionally, this cascaded design ensures efficient computation by progressively reducing the number of candidate windows. Moreover, its hierarchical structure allows real-time performance even on low-power devices.

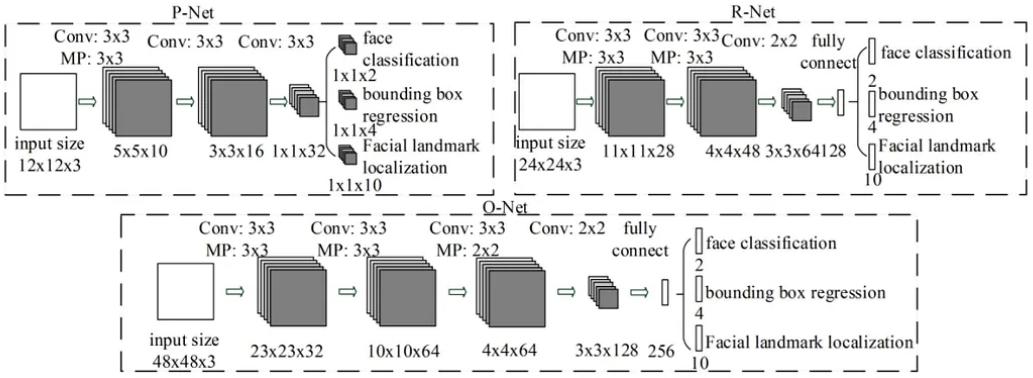


Figure 2.2: Representation of Various Layers of Multi-Task Convolutional Neural Networks (MTCNN) Algorithm [22].

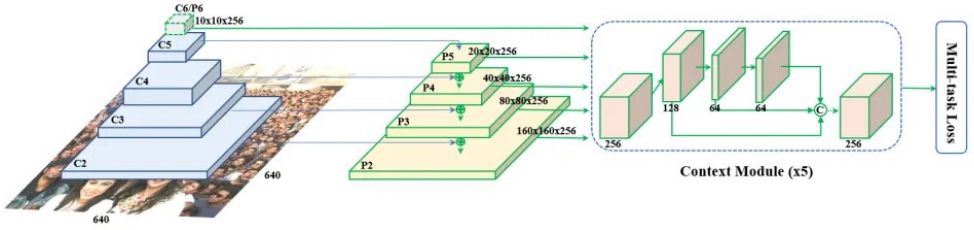


Figure 2.3: Retinaface Algorithm [27].

a single forward pass, making it highly suitable for real-time applications. It employs a feature pyramid network to support hierarchical processing, which allows the model to detect faces across multiple scales. This is particularly beneficial in scenarios involving large variations in face sizes within the same image. To improve detection performance, especially for small or occluded faces, RetinaFace incorporates extra supervision by using manually annotated facial landmarks and leverages self-supervised learning techniques to enhance its feature learning capabilities. RetinaFace achieves high accuracy on benchmark datasets, even under challenging conditions such as crowding or poor lighting.

Its scalable design supports detecting faces of various sizes and can be optimized for specific use cases, including lightweight versions for CPU-based real-time applications. The combination of precision, speed, and flexibility in backbone selection makes RetinaFace a versatile and powerful model for face detection across a range of computer vision tasks and also in many other related tasks. It is better than MTCNN for low-resolution images. RetinaFace architecture is provided in Figure 2.3.

2.4.2.1 Retina - ResNet

The Residual Network, commonly referred to as ResNet, is widely used as a backbone in RetinaFace for face detection tasks. Known for its deep architecture, ResNet introduces residual blocks that effectively mitigate the vanishing gradient problem, enabling the training of very deep networks. Variants such as ResNet-50, ResNet-101, and ResNet-152 differ in depth, with larger versions offering more capacity to learn complex patterns. Within RetinaFace, ResNet functions as a powerful feature extractor, capturing rich and hierarchical representations from input images. These features play a crucial role in enabling the accurate detection of faces and the precise prediction of facial landmarks. The use of ResNet enhances RetinaFace's performance, making it highly reliable in challenging conditions involving varied lighting, occlusions, and different facial orientations.

2.4.2.2 Retina - Slim

The Slim backbone is a lightweight version of the network architecture used in face detection systems such as RetinaFace. Designed to minimize computational demands, it often involves a reduced number of layers or simplified operations when compared to more complex networks like ResNet. Despite having fewer parameters, the Slim backbone is capable of extracting essential features required for accurate face detection. Although it may offer slightly lower accuracy than deeper architectures, it provides a practical trade-off between performance and efficiency. This makes it particularly suitable for real-time applications where processing speed and limited hardware resources are critical constraints. Furthermore, its modular design allows seamless integration into embedded and mobile vision systems, enabling efficient deployment across diverse edge platforms.

2.4.2.3 Retina - MobileFaceNet

MobileFaceNet serves as an efficient and compact backbone for RetinaFace, specifically optimized for speed and low computational cost. Its architecture is built

around depthwise separable convolutions, which significantly reduce the number of parameters compared to traditional convolutional layers. This design enables MobileFaceNet to extract crucial facial features while maintaining a lightweight structure, making it ideal for real-time face detection on devices with limited processing power and hence is majorly used in the IoT where usually small devices are interconnncted as bulk a bulk network. In terms of performance, it strikes a practical balance between accuracy and efficiency. While it may not match the precision of deeper networks like ResNet in highly challenging scenarios, it performs reliably across a wide range of face detection tasks and is particularly effective in applications that require quick response times.

2.5 Face Recognition Models

On the face recognition side of the system, four advanced and widely recognized models are employed, each bringing its own strengths and limitations to the task. While these models may differ in terms of accuracy, speed, and efficiency, they all share a common objective: to generate precise and compact embedding vectors for every detected face. These embeddings serve as numerical representations of facial features, allowing the system to compare and recognize individuals effectively. In the upcoming section, a detailed explanation will be provided for each of these models, including the techniques, architectural choices, and specific methods they use to extract facial embeddings from input images.

2.5.1 AdaFace

Recognition in low-quality face datasets is challenging due to the obscured and degraded facial attributes. Advances in margin-based loss functions have enhanced the discriminability of faces in the embedding space. Previous studies have explored adaptive losses that assign more importance to misclassified (hard) examples. Introducing a new aspect of adaptiveness in the loss function, namely image quality, this method adjusts the strategy to emphasize misclassified samples

based on their image quality. Specifically, the relative importance of easy or hard samples is determined by the sample's image quality. This new loss function emphasizes samples of varying difficulties based on their image quality. It achieves this through an adaptive margin function that approximates image quality with feature norms. Furthermore, this approach enables the model to handle intra-class variations more effectively, especially in challenging conditions such as blur or low illumination. By dynamically adjusting margins according to quality, the network learns more robust and generalized facial embeddings. Consequently, it not only enhances recognition performance on degraded datasets but also improves overall model stability and convergence during training. Extensive experiments

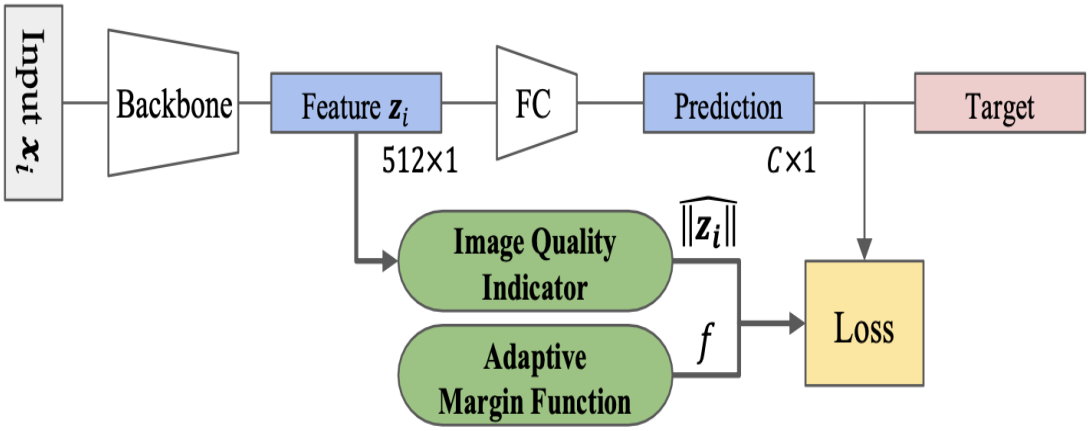


Figure 2.4: AdaFace Block Diagram [28].

show that this method, Ada Face [29] , improves face recognition performance over the state-of-the-art (SoTA) on four datasets: IJB-B, IJB-C, IJB-S, and Tiny Face [30] . Its architecture is given in Figure 2.4. Unlike fixed-margin approaches, AdaFace adapts to individual sample difficulty, making it particularly effective for real-world facial variations. Unlike fixed-margin approaches, AdaFace adapts to individual sample difficulty, making it particularly effective for real-world facial variations.

2.5.2 ArcFace

The field of large-scale face recognition using Deep Convolutional Neural Networks (DCNNs) has grappled with designing loss functions that effectively enhance the

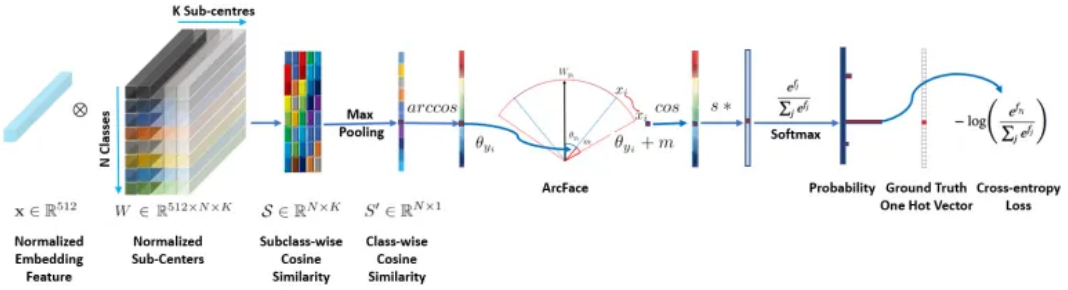


Figure 2.5: ArcFace Block Diagram [31].

discriminative power of learned features. Centre loss achieves intra-class compactness by penalizing the distance between deep features and their corresponding class centers in Euclidean space, as shown in Figure 2.5. Sphere Face, on the other hand, assumes the final fully connected layer's transformation matrix represents class centers in the angular space. It penalizes the angles between deep features and their corresponding weights using a multiplicative approach.Recent research suggests incorporating margins into established loss functions to maximize the separability between face classes. Arcface proposes the Additive Angular Margin Loss (Arcface) [31], specifically designed to generate highly discriminative features for face recognition tasks. Arc Face boasts a clear geometric interpretation due to its direct correspondence with geodesic distance on a hypersphere. The authors conducted a comprehensive evaluation, comparing Arcface against all recent state-of-the-art face recognition methods. The results conclusively demonstrate that Arc Face consistently outperforms existing methods while offering straightforward implementation with minimal computational overhead. [31] Arc face model uses backbone modules of the Resnet family, and it has different variants. Different versions like Resnet 18 and Resnet 50 have been used. The main focus of Arcface is on the loss function [26, 32–34].

2.6 Sub-center Learning and Contrastive Distillation Model

This model is designed to balance high performance with computational efficiency through an advanced deep learning architecture. This model leverages techniques

called Subcenter Learning and Contrastive Distillation Loss to effectively transfer knowledge from a large, pretrained neural network (the Teacher Network) to a smaller, efficient network (the Student Network) [35]. By employing multiple subcenters per class, the model captures intra-class variations more effectively, leading to tighter feature clustering. The contrastive distillation process ensures that the student network preserves both angular margins and relational knowledge from the teacher model. As a result, the system achieves competitive recognition accuracy while significantly reducing inference time and resource consumption. Moreover, this knowledge transfer strategy enhances generalization when deployed across heterogeneous datasets with varying conditions.

2.6.1 Subcenter Learning

In face recognition, a common approach involves training a model to learn a single "center" for each class (e.g., individual). This center represents the ideal features for that class. However, real-world data often suffers from high intra-class variance. This means there can be significant variations in how a single person appears in different images (e.g., due to lighting, pose, resolution). A single center struggles to capture this diversity. Sub-center learning offers a more robust approach. During training, the model learns multiple sub-centers for each class. These sub-centers represent different valid variations within the class. When processing a new image, the model calculates the distance between the image's features and each sub-center. The closest sub-center determines which class the image likely belongs to. LR images often exacerbate intra-class variance. Sub-center learning is particularly beneficial in this scenario. By having multiple sub-centers, the model can effectively capture the diverse ways a person might appear in low-resolution images. This leads to improved recognition accuracy.

2.6.2 Contrastive Distillation Loss

One critical challenge in low-resolution face recognition is the deterioration of performance when dealing with high-resolution images. A novel approach using

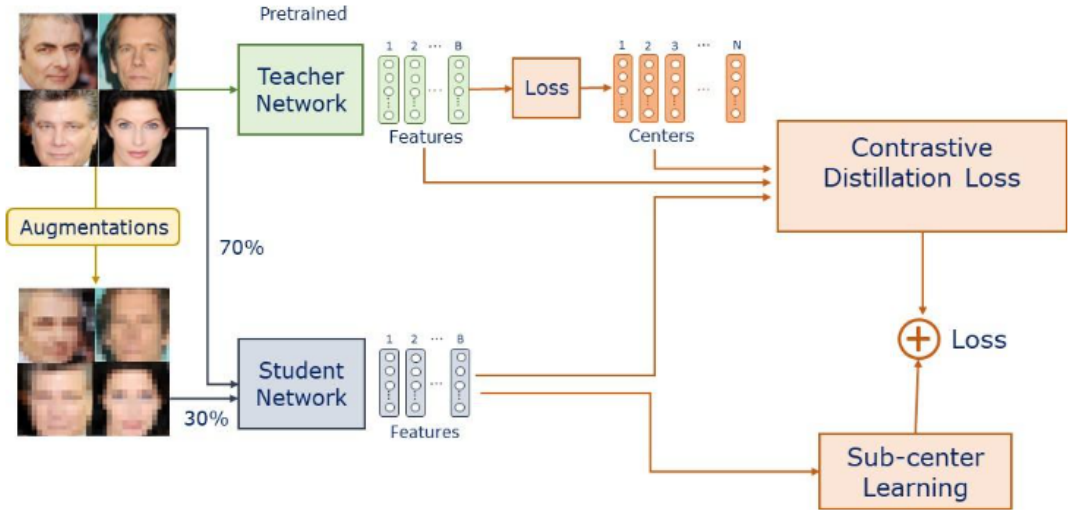


Figure 2.6: Sub-Center and Contrastive Distillation Modeling [35].

contrastive distillation loss to address this issue of disparity between LR and HR features. A key factor contributing to the performance drop is the inherent difference between feature representations extracted from LR and HR images. The student network, trained on LR data, generates low-dimensional feature vectors. In contrast, the teacher network, trained on HR data, produces high-dimensional feature vectors. The proposed contrastive distillation loss aims to bridge this gap by minimizing the distance between the student's LR feature vector and its HR feature vector from the same class.

2.6.3 Noise Contrastive Estimation

This framework allows the model to learn by distinguishing a "query" sample (the LR feature) from a set of "negative" samples (other HR features in the batch).

2.6.4 Margin Parameter

This parameter ensures a larger angular separation between the query and negative samples. This separation is crucial for accurate classification.

- Query This represents the low resolution feature vector extracted by the student network.

- Positive This corresponds to the HR feature vector of the same class as the query, extracted by the teacher network.
- Negatives This set consists of N-1 HR feature vectors from different classes present in the batch.

The distance between the query and positive samples is minimized in contrast to negative samples. By minimizing the contrastive distillation loss, the student network learns to produce LR feature vectors that are closer to their corresponding HR counterparts while maintaining a clear distinction from features belonging to different classes. This alignment between LR and HR features helps the student network perform better when presented with HR images for recognition. Figure 2.6 illustrates subcenter and contrast distillation modeling.

2.7 ViT-Based Face Recognition Models

Vision Transformers (ViTs) offer an attention-based alternative to CNNs by modeling global image contexts. ViTs segment images into patches and process them using Transformer encoders. In the comparison shown in Table 2.1, ViT-Small showed better robustness to corruption and higher accuracy on ImageNet (79%) compared to ResNet50 (76%), although it required more data or smarter pretraining. ViTs are gaining popularity in face recognition due to their scalability and contextual modeling capabilities. A recent study [36] compared ViTs with CNNs, highlighting their better performance in occlusion and pose variation scenarios, although they are computationally more expensive and slower during inference. Furthermore, their ability to model long-range dependencies makes them particularly effective in handling complex facial variations and expressions.

2.7.1 ViT-B/16

Vision Transformer (ViT) models mark a significant paradigm shift in the field of computer vision and face recognition by replacing convolutional operations

with self-attention mechanisms. ViT-B/16, introduced in the original Vision Transformer paper by Dosovitskiy et al. [37], divides an input image into non-overlapping 16×16 patches, which are linearly embedded and passed through a standard Transformer encoder. This architecture leverages multi-head self-attention to model long-range dependencies between facial features, capturing global context more effectively than conventional CNNs. ViT-B/16 has been successfully adopted in face recognition pipelines when combined with advanced loss functions such as ArcFace and AdaFace, which improve feature separability in hyperspherical space [28].

Figure 2.7 shows the general architecture for vision Transformers. Despite its strong representation ability, ViT-B/16 comes with significant computational overhead due to its 86 million parameters and high FLOP requirements, making it more suitable for large-scale server-based deployments rather than real-time or edge applications. Several studies have benchmarked ViT-B/16 on popular face recognition datasets such as LFW, CFP-FP, AgeDB-30, and CPLFW. Results demonstrate that its accuracy is on par with or marginally surpasses that of ResNet-50 when paired with discriminative loss functions, but this comes at the expense of training cost and inference speed.

These trade-offs highlight the need for further optimization when deploying ViT-B/16 in resource-limited environments. Recent research trends focus on hybrid CNN-Transformer architectures that combine local feature extraction with global context modeling to improve efficiency. Additionally, model compression and token pruning strategies are being explored to reduce computational load without sacrificing recognition accuracy.

Table 2.1: Comparison of ResNet vs ViT-Small for Face Recognition [39].

Metric	**ResNet50**	**ViT-Small**
Accuracy (ImageNet)	76%	79% (with DeiT or DINO)
Parameters	25M	22M
Speed	Faster	Slightly Slower
Robustness to corruption	Moderate	Better
Training data needed	Moderate	More or smarter pretraining
Real-world adaptability	Mature	Improving rapidly

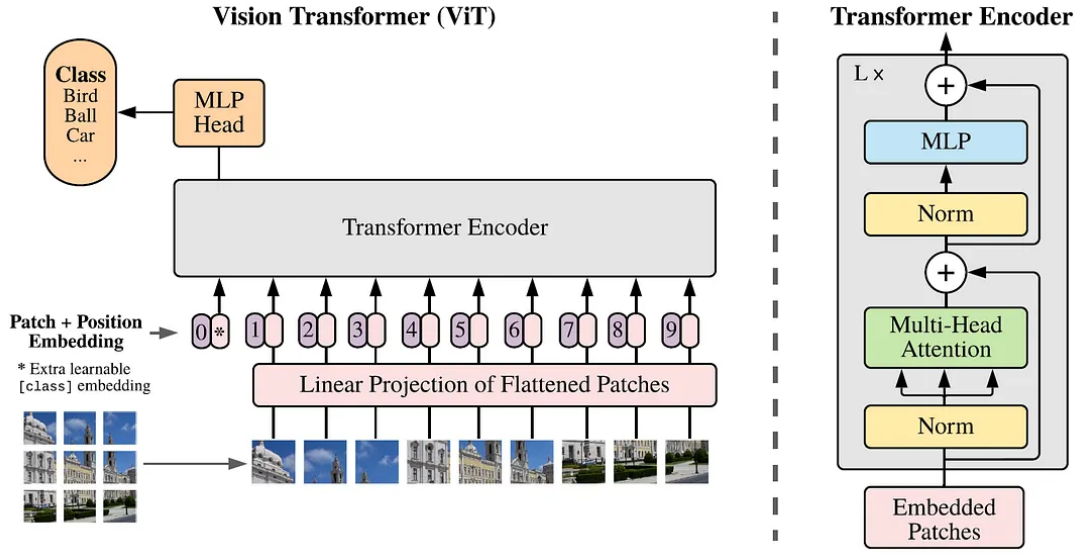


Figure 2.7: Vision Transformers Architecture [38].

2.7.2 ViT-S/16

ViT-S/16 is a smaller variant of the Vision Transformer family designed to reduce the high computational cost and model size of the base and large ViT models. It follows the same architectural principles as ViT-B/16, dividing input images into 16×16 patches and applying Transformer encoders with multi-head self-attention. The primary difference lies in its reduced depth and hidden dimensions, which result in a much lower parameter count (approximately 22 million) and reduced FLOPs. ViT-S/16 is often considered more comparable to mid-scale CNN architectures such as ResNet-50 in terms of parameter size, and it has been investigated in face recognition scenarios with loss functions like ArcFace and AdaFace.

Although it achieves reasonable accuracy on standard datasets such as LFW, AgeDB-30, and CFP-FP, its performance lags behind ResNet-50 when evaluated on challenging cross-age and cross-pose datasets like CALFW and CPLFW. The literature indicates that ViT-S/16 represents a trade-off between accuracy and efficiency: it is lighter and faster than ViT-B/16, making it more practical for deployment, but its reduced representation capacity limits its discriminative power for highly challenging face verification scenarios. Nonetheless, it serves as an effective baseline for exploring lightweight Transformer architectures in real-time face recognition systems.

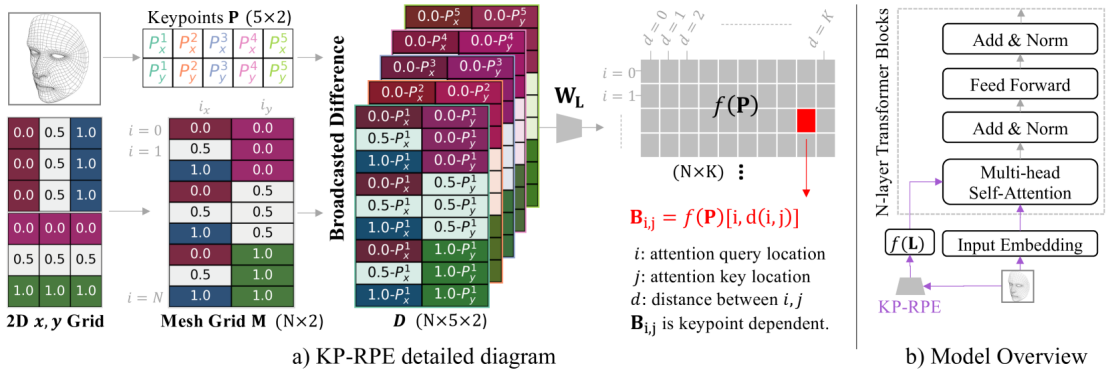


Figure 2.8: The Vision Transformers with Key Point Based Relative Position Encoding (KPRPE) architecture [38].

2.7.3 ViT + KPRPE

ViT with Key Point Relative Positional Encoding (KPRPE), represented as shown in Figure 2.8, presents an enhanced Vision Transformer model designed specifically for face recognition tasks. Traditional ViTs rely on absolute or standard relative positional embeddings, which may not effectively capture fine-grained local relationships critical in faces. KPRPE integrates positional encodings relative to facial landmarks or key points, enabling the Transformer to focus more on discriminative facial regions. The KPRPE-enhanced ViT has been explored in works like CVLFace and AdaFace [28], showing improved accuracy on difficult datasets including CFP-FP and CPLFW. By combining global self-attention with landmark-aware positional encoding, this model captures both global context and local structural details, which are essential for high-accuracy face recognition under large pose and age variations. Despite its superior performance compared to standard ViT-B/16, the model still inherits the high computational and memory costs of transformer-based backbones. It remains primarily suitable for server-grade or offline processing environments.

2.8 Comparative Evaluations from Literature

Despite several evaluations of CNNs and ViTs individually, a surprising lack of unified benchmarks exists that assess both architectures side by side. [40] and [18]

focus solely on CNN-based models, while [36] brings ViTs into the picture but doesn't integrate the evaluation within a common testbed. This fragmented evaluation hampers the development of holistic face recognition solutions applicable to real-world applications.

2.9 Real-World Challenges in Surveillance Scenarios

Face recognition in surveillance involves unique challenges:

- Low-resolution and blurry footage
- Occlusion from masks, glasses, or crowding
- Varying illumination and weather conditions
- Dynamic backgrounds and motion blur
- Occlusion due to sudden pose Variation

In [18], it was observed that CNN-based models degrade significantly under such conditions. ViTs, on the other hand, have shown more resilience due to their non-local attention mechanisms. However, no study yet provides a standard framework that tests both models uniformly on these real-world constraints. Existing datasets often fail to capture temporal variations, low-quality streams, and uncontrolled environmental factors. Therefore, developing evaluation protocols that mimic real surveillance conditions is essential for practical deployment. Bridging this gap would enable a more realistic assessment of model robustness and generalization in the field. Additionally, domain adaptation and unsupervised fine-tuning approaches could help mitigate performance drops across varying camera networks.

2.10 Research Gap Identification

Despite the notable progress in face recognition using both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), existing research still presents

two critical gaps that hinder effective deployment in real-world surveillance scenarios:

- Lack of Comprehensive Comparative Evaluation While both CNN-based and ViT-based models have demonstrated high accuracy on benchmark datasets, there is a lack of a unified, rigorous comparison of their performance under challenging, real-world surveillance conditions characterized by occlusions, motion blur, low resolution, and variable lighting. Most existing evaluations focus only on accuracy and ignore other practical factors such as robustness, model complexity, and inference efficiency.
- Absence of a Modern Unified Framework Popular open-source toolkits like InsightFace are primarily CNN-centric and have not kept pace with the emergence of ViT-based face recognition models. On the other hand, ViT models are supported by modern but isolated frameworks. This fragmentation prevents standardized testing, comparison, and deployment of models across both domains in a single platform.

These gaps highlight the need for a modern, unified evaluation framework that integrates both CNN and ViT-based models, supports consistent preprocessing and detection pipelines, and enables reliable comparison across multiple real-world performance metrics. Addressing this gap is essential to advancing the deployment of face recognition systems in practical surveillance applications.

2.11 Problem Statement

Despite the rapid advancement of deep learning techniques in face recognition, deploying these systems in real-world surveillance environments remains a considerable challenge. Surveillance videos often suffer from low resolution, motion blur, varying illumination, occlusions, and non-frontal facial poses, which significantly degrade the performance of face recognition algorithms. Convolutional Neural Networks (CNNs) have historically dominated the field with architectures

such as ArcFace, AdaFace, and MobileFaceNet, showing strong performance on benchmark datasets. More recently, Vision Transformers (ViTs) have emerged as a promising alternative due to their ability to model global dependencies and achieve high accuracy in vision tasks. However, current literature lacks a comprehensive and fair evaluation of CNN and ViT-based models under practical surveillance conditions. Most comparative studies focus solely on accuracy, often ignoring other crucial deployment factors such as model size, computational complexity, inference time, and robustness to noise and distortions.

This research aims to address the following core problems:

- There is no standardized framework that evaluates face recognition models across both CNN and ViT domains using consistent preprocessing, detection pipelines, and performance metrics.
- Existing evaluations rarely consider real-world conditions present in surveillance footage, leading to over-optimistic conclusions.
- It remains unclear whether the marginal accuracy gains offered by ViT-based models justify their higher computational cost and complexity when compared to efficient CNN-based alternatives.

To bridge this gap, a detailed comparative study is required to evaluate multiple face recognition model combinations on both benchmark datasets and custom surveillance footage. This will help determine the trade-offs between accuracy, efficiency, and practical feasibility, ultimately guiding the selection of optimal models for real-world deployment.

2.12 Summary

This review outlined the evolution of face recognition from traditional methods to CNN and ViT architectures, emphasizing the need for unified, adaptive, and explainable frameworks. Future work should focus on multimodal, energy-efficient, and scalable systems to enhance real-world reliability and ethical deployment.

Chapter 3

Methodology

3.1 Overview

In this section, the methodology used to achieve the thesis's objectives is discussed. All the models and key concepts are explained in the literature review section of the thesis. Four state-of-the-art FR models and four state-of-the-art models for the face detection part are used. Each face detection model is tested against all four FR models, i.e., the combination of Retina Face as detector and ArcFace as recognition model, Retina Face as detector and Adaface as recognition model, and so on, so there are 16 possible combinations. The best-performing combination is selected on the basis of performance to maximize the project objective. The following schematic diagram 3.1 present the software algorithm for the thesis. To start, the face detection models used include Retina Face (with Resnet, Slim, and MobileNet backbones) and MTCNN, known for their high accuracy and efficiency in identifying facial regions within frames. These models detect faces in the input frames extracted from the video feed captured by surveillance cameras. After detection, the faces are subjected to a normalization process to ensure geometric alignment, which is crucial for accurate feature extraction and comparison. The face recognition models, including Arcface, Adaface, Proposed Model, are then employed to generate unique feature embeddings for each detected face. These

embeddings are compared against a database of known faces to determine matches. The combination of face detection and recognition models is evaluated based on metrics such as accuracy, speed, and robustness against variations in lighting, pose, and occlusion.

Additionally, the methodology involves the implementation of a quality assessment module that selects the best possible frame from the video for face recognition, ensuring that the system works effectively even in challenging conditions. The selected frame undergoes preprocessing steps, including resizing, normalization, and alignment, to prepare the frame for feature extraction. Furthermore, the extracted features are compared using similarity metrics such as cosine distance or Euclidean distance to determine identity matching. The system also integrates tracking mechanisms to maintain consistent recognition across multiple frames, reducing redundant computations. To enhance scalability, embeddings are stored dynamically in a Gallery inside the facebank that is updatable.

3.2 Preprocessing

Preprocessing starts with the input to the system. In the facial segment, the input is a video. There is no restriction on the format of videos; in addition, the system has no restriction on the resolution of the video. Both low and high-resolution videos can be tested. As can be seen from the flow chart, the video location is in the face bank. So, it is necessary to ensure that the video on which testing is being performed, must be in the face bank folder within the default project directory. While loading the video from the face bank, it is checked whether the video opens successfully or not. In case there is any error, the user is notified on the spot that the video is not being loaded. Once the video loads, its frames are extracted one by one. An important point here is that all normal videos having moderate resolution contain 30 frames per second, but the human movement is relatively slow, with the exception that someone is using fast displacement objects, like skating shoes, etc. Normally, at different entrances, human movement is such that

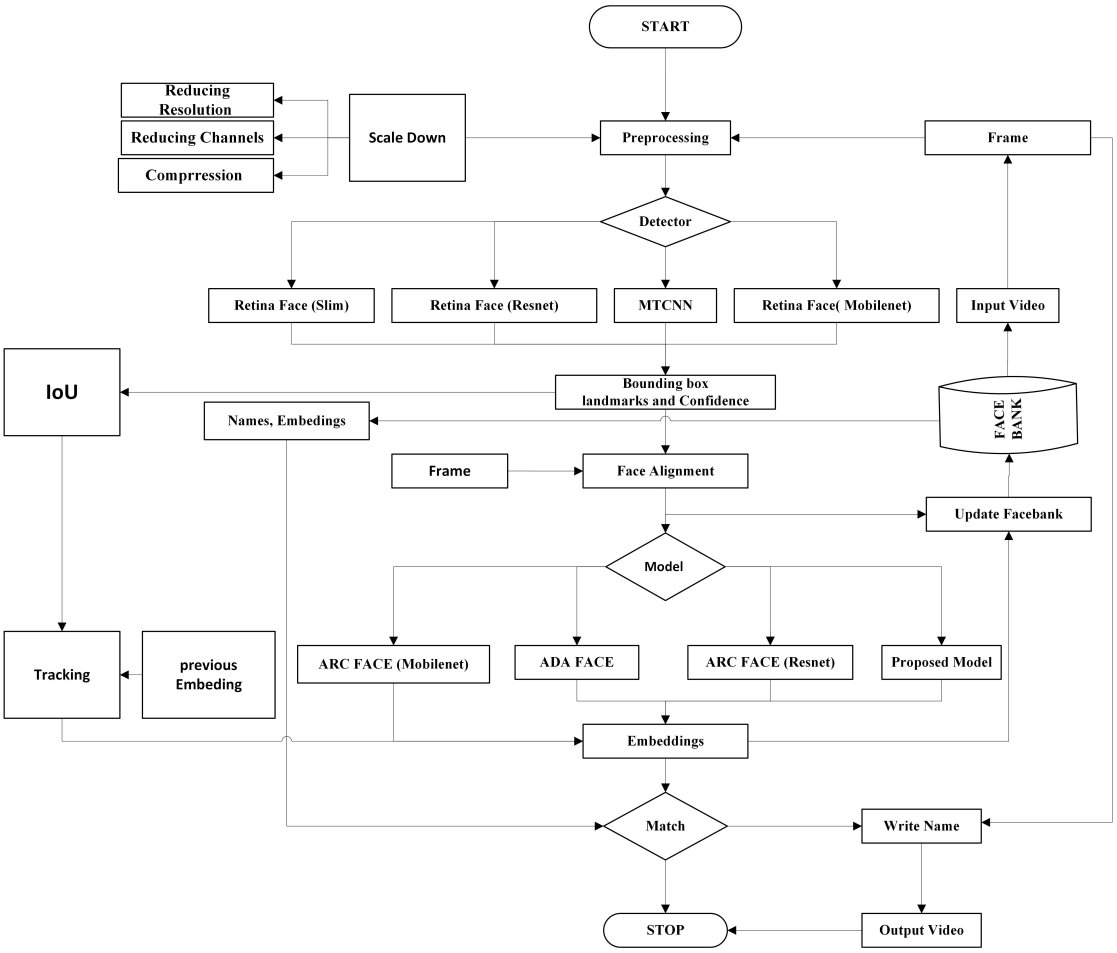


Figure 3.1: Schematic Diagram of the Pipeline

the scenario changes around 2 to 3 seconds. This means that 60 to 90 consecutive frames contain the same information or represent the same individuals. Hence, considering each frame is not necessary to proceed, as usage of all the frames makes the system costly but ensures maximum information transfer. So, there is a trade between the information content and time complexity, which is why every 100th frame is preferred for processing. The selected image goes through preprocessing where it is resized to 112 x 112 pixels, and also normalizations take place.

3.2.1 Face Detection and Alignment

The frame or image is still not ready to be passed to a detection model, as the pipeline comprises multiple detection models. There are two variants of the RetinaFace model and one MTCNN model. The input format varies across these

models: MTCNN requires input images in NumPy array format, whereas RetinaFace variants expect images in PIL image format. The same applies to image size requirements. Therefore, image preprocessing depends on the detection model being used. For MTCNN, the image is resized to 112×112 and then converted into a NumPy array (or ndarray) before being passed to the model. In contrast, for RetinaFace, the image is directly passed to the model after resizing. Most state-of-the-art face detection models follow the same principle, typically divided into two stages: landmark detection and alignment. In MTCNN, both the detection and alignment functions/methods are implemented within a single class. In contrast, for RetinaFace, separate classes are employed for detection and alignment. After minimal preprocessing, the image is passed to the detector method. As mentioned earlier, the required input format varies from model to model. To accommodate this, conditional formatting is applied before passing the image to the detector. Additionally, conversion is handled within the detector if the input format is invalid. If the format is still not correct, the program displays a format error message. Once the image is received in the correct format, the detector network begins processing the image according to the constraints specified in the configuration. The main parameters include the minimum face size, threshold, and non-maximum suppression score. These constraints are discussed in detail in the "RoI Detection Stepwise" section, where each step corresponds to a deep neural network layer. Each layer is explained in depth, along with its effect on the input image at that stage. These layers are common to all three detection models used in the project. There are three primary layers, and each operates on the image based on the specified constraints. This is why the parameters, except for the input image, are provided as a list of three values—one for each layer. This allows for separate thresholds and configurations for each layer. There is always a trade-off between the minimum face size and computational complexity. These parameters are crucial in the face detection process and are typically tuned manually using a trial-and-error method. Their optimal settings also depend on the resolution of the image. For instance, in low-resolution images containing multiple faces, those in the background tend to appear smaller. To detect them, the minimum face size parameter must be set lower, which increases the computational cost.

The image passes through all three layers. At each layer, the bounding boxes and facial landmarks are refined, and regions likely to contain faces are identified. At the final layer's output, two key results are obtained: landmarks and bounding boxes. These are the main outputs returned by the detector function or module. This can also be confirmed from the flowchart, which shows that the outputs of the detection models are landmarks, bounding boxes, and cropped faces or regions of interest (RoIs). So far, the results include landmarks and bounding boxes. The faces are identified and their landmarks located. Landmarks are defined as five facial points representing the x and y coordinates of the left eye, right eye, nose, left corner of the mouth, and right corner of the mouth. The format of these landmarks varies by model. In MTCNN, the landmarks are shaped as a 10×1 vector: the first five elements represent the x-coordinates, and the last five represent the y-coordinates, in the order described above. The landmark format is discussed in more detail in the face alignment section. The second key output from the detector is the bounding box, which provides information about the location of the face within the image. The detector draws a rectangular boundary around the face, including a small offset to ensure a slight margin between the box and the face. The bounding box consists of five values: the x and y coordinates of the upper-left corner of the rectangle, the width and height of the box, and the fifth value is the confidence score.

3.2.2 Alignment and Inference Speed

For real-time applications, it remains a concern that the application response time should be quick enough with minimal lag. Hence, the more complex the algorithm, the more lag the algorithm will have. Similarly, accuracy often has to be compromised to increase speed. With this perspective in mind, the algorithm of alignment established in Part 1 was revised. The earlier version achieved remarkable accuracy for both tiny and distant faces, but the inference time, both on pre-saved video and live feed, was quite low. After a thorough review and research, the following findings were established. With the quality of videos, the number of frames per second (FPS) varies. A video of an average resolution has an

average frame rate of 30, which means that in a second, 30 frames are processed. If this speed is compared with the motion of a person in a video or live feed, the information across frames does not change much. Analysis of different videos showed that consecutive frames in a video often contain the same information. Passing each frame through the detection model and the face recognition model causes large repetition and makes the program computationally extensive. After extensive testing, two solutions were proposed for this problem.

3.2.3 Frame Selection

In this method, certain frames are skipped and only a few frames are allowed to pass through the detection and recognition models. This approach utilizes the face recognition models more efficiently. For example, every 20th or 50th frame can be processed while skipping the others, which speeds up inference since only a subset of frames is processed. Tests on different videos by reading every 20th, 40th, and 60th frame showed that the context can still be captured effectively by processing just 2 or 3 frames per second. Therefore, passing every 20th or 30th frame works well in many scenarios. However, the number of frames to skip is not fixed and should be adjusted according to the specific scenario. For instance, if the video feed comes from a fast-moving or crowded environment, fewer frames should be skipped, for example, processing every 10th or 15th frame, to ensure important moments are not missed, and vice versa in less dynamic settings. Although this method significantly reduces response time, frame skipping has some limitations. One major drawback is that only the frames passed through the detection model are written to the annotated output video. As a result, skipping a large number of frames can produce a very short output video or, in some cases, make the output video difficult to interpret due to missing frames that were skipped.

3.2.4 Tracking

In this method, certain frames can be skipped and allow only a few frames to pass through the detection and recognition models. This approach allows us to

utilize the face recognition models more efficiently. For example, process every 20th or 50th frame while skipping the others, which speeds up inference since only a subset of frames is processed. Tests were performed on different videos by reading every 20th, 40th, and 60th frame, and found that still context can be effectively captured by processing just 2 or 3 frames per second. Therefore, passing every 20th or 30th frame can work well in many scenarios. However, the number of frames to skip is not fixed and should be adjusted according to the specific scenario. For instance, if the video feed comes from a fast-moving or crowded environment, fewer frames should be skipped, for example, processing every 10th or 15th frame, to ensure important moments are not missed, and vice versa in less dynamic settings. Although this method significantly reduces response time, frame skipping has some limitations. One major drawback is that only the frames passed through the detection model are written to the annotated output video.

3.3 Trackers

Trackers are algorithms used for tracking objects, typically in motion. Essentially, tracking involves predicting the location of an object in upcoming frames of a video or live feed. This capability is useful in various applications, such as enabling companies to monitor and track employees across multiple camera views. Two well-known tracking algorithms are SORT (Simple Online and Realtime Tracking) and Deep SORT. In addition to these, custom tracking functions can also be implemented using techniques such as Intersection over Union (IoU). In fact, SORT itself is based on the IoU algorithm for tracking.

3.3.1 SORT

SORT (Simple Online and Real-time Tracker) is a lightweight and efficient object tracking algorithm that works on the tracking-by-detection approach. SORT uses a Kalman Filter and the Hungarian Algorithm to smoothly track detected objects. When integrated into the pipeline, SORT reduced the inference time by almost

50%. It performed well for object tracking in the custom dataset, where most videos involved a person approaching the camera from a certain distance. SORT was tested on more than 20 videos, including scenarios where individual clips of different identities were combined into new videos. In such cases, where different individuals approached the camera one by one, the algorithm maintained stable tracking. However, in more challenging conditions, such as when multiple persons approached the camera simultaneously, tracking performance degraded. Although SORT was assigning IDs correctly, the inference time reduction was not significant. Debugging and repeated testing revealed that with multiple people, the detected faces became smaller, and the detections provided to SORT were inconsistent. As SORT heavily depends on detections, instability in bounding boxes made the tracker ineffective. For example, when a person was facing the camera, the face detection module provided stable detections, but when the person bent, rotated, or became occluded, detections became null. Feeding these null detections to SORT caused it to lose track, and once the face reappeared, SORT assigned a new ID. Frequent repetitions of this phenomenon led to unreliable tracking. Although temporary logic was implemented to mitigate this issue, frequent ID reassignments persisted, creating identification problems. In summary, SORT is extremely fast but sensitive to unstable detections, particularly under occlusion or temporary disappearance of the face. This makes SORT suitable for scenarios requiring high-speed tracking, but less effective in environments where face visibility fluctuates. SORT performs reliably with YOLO-based object detection, as YOLO provides continuous detections, unlike face detection modules, where instability is common. These enhancements enable more consistent identity tracking across frames, even under partial occlusions or missed detections.

3.3.2 Deep SORT

To address the limitations of SORT, Deep SORT was integrated into the pipeline and evaluated under the same scenarios. Deep SORT demonstrated superior performance in terms of ID assignments, as deep SORT was able to track faces across multiple frames while assigning new IDs less frequently. This resulted in more

stable tracking even under partial occlusions. Despite its accuracy advantages, Deep SORT did not improve inference speed. The root cause lies in the difference between the two algorithms. Unlike SORT, Deep SORT incorporates feature embeddings to calculate similarity between detections and existing tracks, similar to a face recognition model. While this embedding-based approach improved accuracy and stability, deep sort also introduced significant computational overhead. Since embeddings were already being computed in the face recognition module, the additional embeddings required by Deep SORT further increased pipeline complexity, thus slowing down inference.

3.3.3 Proposed Tracking Algorithm

Extensive testing of SORT and Deep SORT highlighted that neither could be directly applied for efficient face tracking. Therefore, a specialized tracking algorithm was designed to incorporate the strengths of both approaches. An additional literature review of SORT and Deep SORT was conducted, and based on the findings, a hybrid solution was developed. The IoU-based association method from SORT was combined with embeddings from the face recognition model, inspired by Deep SORT.

The key advantage of this design lies in efficiency: since embeddings were already being computed by the face recognition (FR) model, no additional overhead was introduced. This allowed stable ID generation without frequent reassignments while maintaining fast inference. The proposed tracker proved to be both more efficient and faster than SORT and Deep SORT, while also providing consistent accuracy under volatile face detections. As SORT and Deep SORT are general-purpose object trackers, their performance heavily depends on stable inputs. Face detection, however, often produces volatile outputs due to occlusion, poor lighting, or pose variations, leading to problems such as name swapping. The proposed algorithm addressed these challenges by leveraging FR-based embeddings for re-identification, which stabilized tracking without increasing computational complexity. The new method was tested under multiple conditions and consistently

achieved reliable tracking, establishing it as a more optimized and specialized tracker for face recognition pipelines.

3.4 Identity Swapping Causes and Solutions

In cases where multiple people appear in a single frame, face-detection models like MTCNN and RetinaFace detect the faces of each individual. However, these detections, regardless of the strength of the face detection model, are highly volatile. There are multiple reasons why face detection models sometimes fail to capture faces, resulting in null values for bounding boxes and landmarks. This phenomenon can occur very quickly, such that even in consecutive frames, the detections are inconsistent despite having the same number of individuals in each frame. This phenomena typically happens when a person suddenly turns around, bends, or becomes occluded due to some disturbance. In these conditions, the confidence score of the detection falls below the minimum threshold required to consider the face a valid face. When this phenomenon occurs frequently, traditional trackers like SORT and Deep SORT start assigning new IDs rapidly. As a result, the same person in a video may receive more than 30 unique IDs in less than half a second of footage. This creates confusion in the matching function that determines whether an incoming track is new or existing. Consequently, the tracker starts swapping the identities of matched tracks. For example, if there are three persons named A, B, and C, then A should consistently be tracked as A. However, due to frequent ID reassignments, A may be tracked as B, or B may be tracked as C, and so on. This issue is known as *swapping*. Swapping does not occur under stable conditions where face detections are continuous. Similarly, videos where only a single person appears per frame do not exhibit name swapping or frequent ID assignments. Therefore, it is crucial to extensively test tracking algorithms in the context of face recognition, ensuring they are evaluated against all possible edge cases that could lead to failure before finalizing a solution. Comprehensive evaluation under diverse motion patterns and occlusion scenarios can further help identify and mitigate such ID-switching issues effectively.

3.5 Face Bank and Resolution Problem

In part 1 of the thesis, 16 different pairs of face recognition and face detection algorithms were tested, and the results were recorded in both graphical and tabular formats to facilitate analysis. After a comprehensive review, it was found that all combinations involving MTCNN and face recognition models performed poorly, whereas combinations involving RetinaFace and face recognition models showed promising results. The testing was conducted using a custom dataset collected during part 1. All videos used in testing were recorded using cameras with similar specifications, resulting in videos of comparable quality, with average resolutions ranging from 1200 to 1400. Additionally, images of all 48 students were captured using high-resolution smartphone cameras, with the backgrounds removed. Thus, the dataset consisted of high-resolution images in the face bank and medium-resolution videos. With this information in hand, the second part of the project focused on optimizing MTCNN for faster inference with acceptable accuracy and enhancing the accuracy of RetinaFace while reducing its detection time. Another important objective was to evaluate the generalizability of the pipeline by testing pipeline on videos and corresponding face banks outside the custom dataset. To achieve these goals, the entire pipeline was divided into components: detection alignment, detection plus alignment, embedding generation and distance calculation, and the recognition module as a whole.

After segmentation, the FPS (frames per second) was calculated at the end of each section, and different FPS values were displayed dynamically on the output. This approach helped identify the sections consuming more time and revealed potential bottlenecks. Initially, it was assumed that the recognition part, where embeddings are generated and compared to the face bank was the most time-consuming. However, detailed analysis revealed that the detection stage required more processing time than recognition. This pattern was consistent across all 16 combinations, indicating that both RetinaFace and MTCNN were the primary contributors to delays. Hence, optimizing the detection pipeline or employing lightweight detectors could significantly improve the overall real-time performance of the system.

3.5.1 Key Observations

- Identity swapping occurred in certain videos.
- The performance of MTCNN was consistently the worst across multiple videos.
- While using RetinaFace, some individuals were not recognized despite having their images in the face bank, whereas others were correctly recognized.
- Placing multiple images in the face bank improved recognition performance when using RetinaFace.
- Recognition performance was not affected by placing multiple images for a single identity in the face bank when using MTCNN.
- The generalizability of the pipeline was limited.

These observations highlighted limitations in the pipeline that required attention to achieve a more robust and generalized system. The first observation and its solution had already been discussed earlier. The second was unsurprising, as MTCNN consistently underperformed compared to RetinaFace; hence, no further effort was invested in improving MTCNN's accuracy.

The third observation was unexpected, given that RetinaFace had produced promising results in part 1. This issue was identified during the testing of a video featuring three individuals seated on chairs. The video resolution was moderate—neither very high nor very low—so the input video could not be categorized as low-resolution. For this test, the face bank was updated with images and names of the individuals, captured using smartphone rear cameras. These images, unlike earlier ones, were not preprocessed (backgrounds were not removed). Despite being high-resolution, these images failed to produce correct recognition results. Additional testing with varied parameters on both the detection and recognition sides did not resolve the issue. Similar behavior was observed in other videos containing multiple people in a single frame. Upon further debugging—ensuring embeddings and name files were generated, cross-checking sizes, and validating

face bank updates and discovered that some individuals were consistently not being recognized, even across different videos. Adding alternative images of the same individuals to the face bank, including cropped versions, resolved the problem for some cases, suggesting the issue was related to face bank updates rather than the pipeline itself. Printing the contents of the `names.npy` file revealed that some names were missing despite the presence of corresponding images, confirming that the face bank was not being updated properly. For example, although 56 images were present in the face bank, only 35 embeddings were being generated. Further testing showed variability depending on the RetinaFace backbone: RetinaFace (Slim) generated only 29 embeddings, while RetinaFace (ResNet) generated 35. In contrast, switching the detector to MTCNN enabled all 56 embeddings to be generated successfully, despite MTCNN's otherwise poor detection performance. A temporary solution was to use MTCNN for face bank updates and RetinaFace for detection in frames. While this resolved the issue, the use of three models simultaneously was not feasible in the long run. A deeper analysis of RetinaFace revealed that RetinaFace had been trained primarily on low-resolution images and optimized for detecting small faces, whereas face recognition models were optimized for high-resolution inputs.

This mismatch explained the observed failures: RetinaFace struggled with high-resolution, sharp but small-sized face bank images, and occasionally misidentified background faces. Attempts to solve the issue by adjusting thresholds within RetinaFace (confidence, NMS, and visualization) improved embedding generation to about 90% but caused significant degradation in inference performance. Frames were skipped, detection slowed, and accuracy dropped to levels comparable to MTCNN.

3.5.2 Final Insight

Further testing confirmed that:

- Thresholds optimized for face bank updating degraded detection in video frames.

- Thresholds optimized for video detection prevented proper face bank updates.

These findings indicated that RetinaFace could only perform effectively when the resolution of inputs was consistent across both the face bank and test videos. Since most CCTV and surveillance footage is low-resolution, the solution was to align the resolution of the face bank accordingly. Replacing high-resolution images with medium-resolution ones resolved the problem. With this adjustment, recognition accuracy improved significantly in all scenarios, and the issues related to face bank updates were fully resolved.

3.6 Inference Speed Optimization

The pipeline of the project was divided into modules, and the FPS of each module was calculated. This division was done to identify which modules were responsible for longer processing times. During this analysis, it was found that the face detectors were consuming more time than any other module in the pipeline. This was evident when the alignment module returned the lowest FPS, indicating that detection was the most time-consuming part, as a lower FPS corresponds to higher processing time. The alignment function consists of three main components.

Face detection

Landmarks detection and angle calculation

Geometric transformation

3.6.1 Face Detection

Face detection has been explained in detail earlier in this report. To summarize, a frame from a live feed or a prerecorded video is read using a suitable tool like OpenCV. Bounding boxes are then drawn around the detected faces, and the

region of interest (RoI) is cropped. This process is applied to all frames, and each person appearing in a frame undergoes this cropping process if their image meets the detection threshold. Images that fall below this threshold are not considered valid detections, as it becomes difficult to accurately detect landmarks or facial points in such small or unclear images.

3.6.2 Landmarks Detection and Angle Calculation

Angle calculation is performed using reference landmarks and source landmarks. In this step, facial points, commonly referred to as landmarks, are compared to predefined reference landmarks to estimate how much the source image needs to be rotated in order to produce an aligned image. The goal is for the facial points in the rotated image to closely match the reference facial points. These facial points usually include the x and y coordinates of the left and right eyes, the nose, and the left and right corners of the mouth. The reference landmarks are the same set of coordinates, but based on a standard 112×112 aligned image. The distance between each facial point in the source image and its corresponding reference point is calculated, and from these differences, the rotation angle is determined. The number of landmarks varies from model to model. For instance, MTCNN returns five landmarks in the form of x, y coordinate pairs. In contrast, RetinaFace returns 10 values, the first five x coordinates followed by five y coordinates, which must be manually paired before use. Moreover, the number of landmarks is not fixed; landmarks can exceed five. For example, RetinaFace supports mesh-based landmark detection, and models like DLIB can return 68 landmarks, including the jawline and eyebrows. These extensive sets are typically used to create a facial mesh resembling a mask. The choice of the number of landmarks depends on user requirements. However, using more landmarks increases the complexity of the detector, which in turn decreases detection speed. To maintain uniformity in the project, five landmark pairs were used, as MTCNN does not support mesh-type landmarks. Since the minimum required number of landmarks for effective face alignment was already being used, there was no margin for further reducing landmarks in this module to optimize inference speed. The third component of face

alignment is geometric transformation. In this stage, the distances between landmarks and the rotation angle are used to compute a transformation matrix. This matrix is then applied to the original image, resulting in a transformed (aligned) image whose landmarks match the reference landmarks. There are several types of transformations used in face alignment, including:

3.6.3 Affine Transformation

Affine Transformation is the linear mapping method that preserves points, straight lines, and planes. A minimum of three landmarks is required for this transformation. The degree of freedom in an affine transformation is 6. These are the important steps involved in an affine transformation.It allows scaling, rotation, translation, and shearing operations while maintaining the geometric structure of the image. This transformation is widely used in face alignment tasks to ensure that key facial features such as eyes, nose, and mouth are positioned consistently across different samples.

Scaling

Rotation

Translation

Shearing The degree of freedom corresponds to the number of independent parameters that define the transformation. As is known, the affine transformation matrix is a 2×3 matrix where a, b, c, and d are the values used for scaling, rotation, and shearing. The other two parameters, t_x and t_y, are used for linear transformation.

3.6.4 Similarity Transformation

Similarity transformation is a special case of an affine transformation. Similarity transformation typically preserves shapes, angles, and aspect ratios. Unlike an

affine transformation, a similarity transformation does not involve shear calculation, making it a lighter and more efficient transformation. Additionally, Affine Transformation offers uniform scaling, which is not the case with an affine transformation. The degrees of freedom in similarity transformation are four, meaning only four independent variables control the transformation. Although an Affine Transformation typically requires evaluation of five landmarks, the minimal requirement is two to three landmarks, similar to an affine transformation. However, for higher accuracy, five landmarks are usually used in both cases. Landmarks used for the affine transformation should be non-collinear. Comparing the two transformations, the similarity transformation is considered better for face recognition, especially when the objective is to increase inference speed while maintaining accuracy. The project pipeline includes an option for an additional variant of the affine transformation. All transformations were tested and the similarity transformation was found to outperform the affine transformation, both in accuracy and inference time.

Therefore, it was decided to switch from using an affine transformation with five landmarks (as used previously in part 1) to a similarity transformation. This improvement stems from the fact that similarity transformation retains geometric consistency across varying image scales and orientations. It also minimizes distortions that often arise during affine warping, resulting in more stable facial alignment. Furthermore, its reduced computational cost makes it ideal for real-time face recognition systems deployed on limited hardware.

3.7 Impact on Inference

The inference time did not change significantly since the landmarks detection and transformation parts had already been thoroughly researched. However, replacing the affine transformation with the similarity transformation did result in a slight increase in inference speed. Now, only the face detection part remains to be reviewed. The latter two parts—landmarks detection and transformation—were examined first because they had not been explored in detail during part 1, and the

aim was to identify any potential optimizations there. Regarding face detection, the challenge was that the image could not be altered much, as any part of the image could contain a face. Additionally, resizing the image increases the risk of missing tiny faces. To better understand the logic and functions used in face detection, debugging was carried out on that part of the pipeline. Other detection algorithms beyond face detection were also researched. During this process, the problem was discussed with the project supervisor, Dr. Imtiaz Taj. After reviewing the issue, he suggested reducing the image resolution when passing the frame through both face detection and landmarks detection, then mapping the detected landmarks back to the original image. Once the landmarks were mapped, the original image would be passed to the transformation function. Upon researching various face detection algorithms implemented by different contributors, including RetinaFace and InsightFace, this approach was missing in their methods. Implementation steps for scaling and mapping back are as under:

- Downscaling the input image
- Face detection and landmark detection
- Mapping the detection back to the original image

3.7.1 Downscaling the Input Image

The image can be downscaled in several ways. One method is to reduce its resolution by decreasing its dimensions while maintaining the aspect ratio. Another approach is to reduce its rank, which refers to lowering the number of dimensions. The depth of the image, or the number of channels, is typically 3, either in RGB or BGR format.

3.7.2 Mapping the Detections Back to the Original Image

Mapping the landmarks returned from the detection phase to the original image is known as landmark mapping. This is the process of locating the obtained

landmarks within the original image. Once these landmarks are mapped back, the original image is then ready to be passed to the transformation step, where the RoI is aligned according to the reference landmarks. This alignment ensures that the image is in a suitable orientation and is ready to be fed into the face recognition model, where its embedding will be generated and later compared with other embeddings.

3.7.3 Implementation of Proposed Logic

After thorough research, the results were that three factors can have a slightly greater impact on the resolution of the image. By controlling these factors, the input image or frame can be effectively downscaled while keeping the necessary information intact for the face detection and landmark detection processes. Excessive downscaling causes a significant loss of image information. Therefore, the downscaling values of the three selected factors were set to an optimal level that ensures maximum inference speed while preserving sufficient information for accurate detection.

Below are the three factors chosen to modify for image downscaling:

1. Resizing
2. Reducing Number of Channels
3. Compression

3.7.3.1 Resizing

The size of an image corresponds to its width and height in pixels, often referred to as the resolution of the image. A high-resolution image has larger width and height values, while a low-resolution image has smaller dimensions. Resizing an image is a computationally intensive process that involves interpolation and, in some cases, regeneration—predicting missing pixel values, which can be complex. However, passing a resized image to the face detection and landmark detection

modules can significantly reduce inference time. Input images were resized and passed to the detection components. Initially, a test video was used in which neither the width nor the height of the frames exceeded 1500 pixels. Using a hit-and-trial approach, various resizing configurations were tested: 400×400, 600×600, 700×700, and more.

Images resized to below 400×400 yielded poor detection results due to the loss of crucial information. Resizing between 400×400 and 600×600 produced optimal results in terms of both accuracy and inference time. On average, inference time was reduced by approximately 60% across different combinations of face recognition (FR) and face detection (FD) models. For instance, a video previously requiring 10 minutes for processing could now be processed in 4 to 4.5 minutes, or even faster in some cases. This was a dramatic speed improvement. Combined with earlier optimizations (like tracking algorithms, which reduced inference time by 50%), the pipeline achieved a total inference time reduction of up to 75%. Initial thoughts were that the quality of the output video would degrade, and a drop in accuracy would occur. However, the results defied expectations: not only did accuracy improve, but the output video also became more stable. This was especially true for RetinaFace, as these changes had not yet been tested with MTCNN. The improvement was attributed to RetinaFace being trained on low-resolution images, making it inherently better suited for such inputs.

Previously, the high-resolution input images degraded during internal processing, becoming noisy and causing poor embedding generation, ultimately leading to inaccurate recognition. Reducing the input resolution helped mitigate this issue by improving both accuracy and inference speed. To further validate these findings, RetinaFace was tested on various videos with differing resolutions. The optimized resizing worked well across most scenarios except when the frames contained a large number of individuals, resulting in smaller faces. One such case involved a frame with a resolution of 1400×2500, where resizing the image to 400 pixels (while maintaining the aspect ratio) led to excessive downscaling and many missed detections. This highlighted a limitation of the approach. To address this, a classification mechanism was introduced that divides frames into two groups:

- Moderate-resolution images (700×700 to 1500×1500) should be resized to approximately 400×400, maintaining the original aspect ratio.
- High-resolution images (where either the width or height exceeds 1500 pixels) should be resized to 700×700, which proved optimal after multiple iterations.

This adaptive resizing approach ensures that both high and moderate resolution images are efficiently downscaled without losing critical information. It's crucial to maintain the original aspect ratio during resizing. Any deviation can distort the image—causing contraction in one direction and elongation in another—thereby affecting the detection quality.

In summary, downscaling images by reducing their resolution can greatly accelerate inference time—up to 50 times faster—without requiring regeneration, since the original image is still available. Maintaining the aspect ratio is essential to preserving the integrity of visual information. Adaptive resizing based on input resolution not only improves speed but also enhances detection accuracy, particularly when using RetinaFace, which performs best with lower-resolution images.

3.7.3.2 Reducing Depth

The depth of an image refers to the number of color channels the image contains. Most commonly, images are represented in either the BGR (Blue, Green, Red) or RGB (Red, Green, Blue) format, where each channel corresponds to one of the three primary colors. RGB is the most widely used format in image processing and computer vision. Each channel has intensity values typically ranging from 0 to 255. By varying these values across the three channels, a wide spectrum of colors can be produced. Increasing the number of channels (depth) increases the complexity of the image. A higher-complexity image requires more computation and processing time. However, for tasks like face detection and landmark detection, this color information is generally not necessary. These tasks rely mainly on structural features such as edges, corners, and intensity gradients rather than color. This observation suggests that full-color (RGB or BGR) images are not

essential for accurate detection. A grayscale image, which has only one channel and thus a depth of 1, often provides all the necessary information for detection tasks. By converting images from three channels to one (i.e., from RGB/BGR to grayscale), the image depth can be effectively reduced, thereby simplifying the data and enabling faster processing. Grayscale images were tested as input to the detection models, and found that detection time decreased. However, the improvement in inference speed from reducing depth was less significant than the speed-up gained from resizing. Nevertheless, reducing image depth does contribute to a performance boost and complements other downscaling techniques. In addition, grayscale conversion minimizes the impact of color variations caused by lighting conditions, making the model more robust in uncontrolled environments.

3.7.3.3 Compression

Compression involves the permanent removal of data from an image. JPEG compression is commonly used to reduce the size of an image on disk. However, compression does not reduce the size of the image in terms of length and width, meaning that the compressed image retains the same dimensions as the original image. As a result, compression does not contribute to reducing detection time, since the processing is based on the image's dimensions, not its file size. Compression was applied, and the detector was tested on various inputs. Observations were that compression does not lead to any improvement in inference speed. Although image compression is useful for efficient storage and data transfer, but does not enhance the performance of face or landmark detection during inference.

3.8 Feature Extraction

Feature extraction aims to capture meaningful and distinctive representations from the aligned face region that are robust to variations in lighting, pose, and expression [41]. Several approaches exist, ranging from traditional handcrafted methods to modern deep learning techniques.

In this work, all the feature extractors employed are based on the *softmax* classification framework. Given an input feature vector $\mathbf{x}_i$ and class weight vectors $\mathbf{W}_j$, the conventional softmax function is defined as:

$$P(y_i = j \mid \mathbf{x}_i) = \frac{\exp(\mathbf{W}_j^\top \mathbf{x}_i + b_j)}{\sum_{k=1}^{C} \exp(\mathbf{W}_k^\top \mathbf{x}_i + b_k)}, \tag{3.1}$$

where C is the number of classes, and b_j is the bias term for class j.

To enhance discriminative power, *weight normalization* is applied, where each weight vector $\mathbf{W}_j$ and feature vector $\mathbf{x}_i$ is normalized as:

$$\mathbf{W}_j \leftarrow \frac{\mathbf{W}_j}{\|\mathbf{W}_j\|}, \quad \mathbf{x}_i \leftarrow \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}. \tag{3.2}$$

This normalization causes the dot product $\mathbf{W}_j^\top \mathbf{x}_i$ to depend solely on the angle θ_j between the vectors, removing magnitude effects.

The resulting *normalized softmax* function becomes quit easier to understand and is given as under:

$$P(y_i = j \mid \mathbf{x}_i) = \frac{\exp(s \cos \theta_j)}{\sum_{k=1}^{C} \exp(s \cos \theta_k)}, \tag{3.3}$$

where s is a scaling factor controlling the distribution's concentration. This angular formulation forms the foundation for many modern face recognition loss functions, enabling better separation between classes in the feature space.

3.8.1 State-of-the-Art Feature Extractors

Several state-of-the-art face recognition models were employed in this project. Each of these models builds upon the normalized softmax foundation while introducing innovations in their loss functions or training strategies. These models include variants such as ArcFace, CosFace, and AdaFace, each designed to enhance angular margin optimization and discriminative feature learning. By refining how inter-class separation and intra-class compactness are handled, these methods significantly improve recognition accuracy across diverse datasets. Additionally, some

models leverage knowledge distillation and subcenter learning to achieve better generalization with reduced computational cost. Collectively, these advancements contribute to more stable convergence, faster inference, and superior performance under real-world surveillance conditions.

3.8.1.1 AdaFace

AdaFace introduces a novel adaptive margin loss that considers image quality when learning embeddings. AdaFace emphasizes hard examples more if their image quality is poor, improving robustness on degraded images. The method estimates image quality using feature norms and adjusts training dynamically. AdaFace outperforms previous models on IJB-B, IJB-C, IJB-S, and TinyFace datasets. This adaptive strategy enables the model to effectively balance learning between high- and low-quality samples, resulting in stronger generalization across diverse datasets. Unlike fixed-margin methods, AdaFace dynamically refines its decision boundaries based on the difficulty of each sample during training. This continuous adaptation ensures that both easy and challenging samples contribute meaningfully to the embedding space. Consequently, AdaFace achieves state-of-the-art accuracy in complex, real-world face verification and surveillance environments. It not only demonstrates faster convergence and improved training stability but also maintains high discriminative power under varying lighting, pose, and occlusion conditions. Overall, its adaptive margin mechanism establishes a more resilient and context-aware feature learning process.

The AdaptiveFace loss [29] is defined as:

$$f(\theta_j, m)_{\text{AdaFace}} = \begin{cases} s(\cos(\theta_j + g_{\text{angle}}) - g_{\text{add}}), & j = y_i, \\ s\cos\theta_j, & j \neq y_i. \end{cases} \tag{3.4}$$

Here, m is an adaptive margin that varies based on feature norm, and s is the scaling factor. This adaptive margin dynamically adjusts the decision boundary according to the feature quality, allowing the model to treat high- and low-quality

samples differently. As a result, AdaFace achieves better generalization and robustness in unconstrained face recognition scenarios.

3.8.1.2 ArcFace

ArcFace introduces an Additive Angular Margin Loss for learning highly discriminative face features. ArcFace operates in angular space with a clear geometric interpretation on a hypersphere. ArcFace uses ResNet backbones (e.g., ResNet-18, ResNet-50) and consistently outperforms other models across ten major benchmarks. The model is computationally efficient and easy to implement. The ArcFace loss [31] is defined as:

$$\mathcal{L}_{\text{ArcFace}} = -\log \frac{e^{s\cdot\cos(\theta_y+m)}}{e^{s\cdot\cos(\theta_y+m)} + \sum_{j\neq y} e^{s\cdot\cos(\theta_j)}} \tag{3.5}$$

This adds an angular margin m to enhance class separability while preserving intra-class compactness on a hypersphere.

3.8.1.3 Sub-center Learning with Distillation Loss

This model combines sub-center learning with a distillation loss [42] to enhance recognition across image resolutions. This model uses a teacher-student framework where the teacher network is trained on high-resolution images and the student on low-resolution ones. Both networks share the same backbone (ResNet-50), enabling the student to mimic high-quality embeddings. This improves performance under resolution mismatch conditions. Furthermore, the integration of sub-centers reduces intra-class variation by assigning multiple representative centers per class, leading to more compact feature clusters. This hybrid approach not only improves cross-resolution generalization but also enhances robustness in unconstrained and low-quality facial imagery.

$$\mathcal{L}_{\text{ad}} = \sum_{I'\in\mathcal{I}_s} \sum_{f\in\{l,m,h\}} \left\| \frac{A(F_t^f(I))}{\|A(F_t^f(I))\|_2} - \frac{A(F_s^f(I'))}{\|A(F_s^f(I'))\|_2} \right\|_2^2 \tag{3.6}$$

The overall attention-guided distillation loss combines attention-based distillation loss, deep feature distillation loss, and cross-entropy loss function, given by [43]:

$$L_{agd} = L_{cls} + \lambda_1 L_{dfd} + \lambda_2 L_{dad} \tag{3.7}$$

The cross-entropy loss function, denoted by L_{cls}, is adopted as the classification loss for the student network. The subscripts in L_{agd}, L_{dfd}, and L_{dad} correspond to the attention-guided distillation, deep feature distillation, and attention-based distillation losses, respectively. λ_1 and λ_2 are auxiliary weights of the loss function.

3.8.2 Integration into the Face Bank

So far, face detection has been completed, resulting in the aligned (or wrapped) face. As mentioned earlier in the flow chart **??**, the face bank contains multiple entries. The face bank stores aligned faces, which can either be manually added or captured by a camera connected to the program. The project supports both video recording and image capture by specifying the name of the person whose picture or video is being recorded. The program creates a folder named after the specified person inside the face bank directory and saves the captured images there. Images of various individuals were manually placed in the face bank. The wrapped image obtained from the face detection step is then compared against all faces stored in the face bank. If a match is found, the corresponding name from the face bank is assigned to the wrapped face, indicating recognition. Otherwise, a default label "unknown" is assigned to the face. This process, called face recognition, involves complex computer science techniques. Recognition is performed using pre-trained models such as AdaFace, ArcFace, CosFace, and others. In this project, four state-of-the-art face recognition models were used. The architecture and details of these models were provided just before the face detection section. These models are based on deep neural networks, including advanced convolutional neural networks. While the input and output formats of all models are the same, each model processes the input differently. The objective of these models is to extract features that maximize the model's discriminative power. This is achieved through various

techniques, such as loss function optimization. The input image is processed by a selected or user-configured face recognition model, which extracts features and generates an embedding for each image. However, to recognize faces, reference embeddings are needed to compare against. Therefore, updating the face bank with embeddings is necessary.

3.8.3 Selection of Backbone Network

The second step is feature extraction, which is done by passing the image through a backbone network. A backbone network is a multilayer neural network used specifically to extract meaningful features from the input image. AdaFace employs a deep convolutional neural network (CNN), often utilizing ResNet variants. In this project, ResNet-18 and ResNet-50 variants were used for AdaFace. Similarly, ArcFace also uses ResNet variants but with a higher number of layers, sometimes up to 150. For ArcFace, ResNet-50 and ResNet-100 models were tested. ArcFace also occasionally uses squeezed ResNet blocks, known as IR-SE blocks, to improve performance. MobileFaceNet, the third model, employs a lightweight CNN architecture designed specifically for mobile and edge devices. Mobilenet is inspired by MobileNetV2 and focuses on efficiency and speed while maintaining good accuracy. The fourth model is the proposed model. The proposed backbone integrates both convolutional and transformer-based modules to balance efficiency and representational power. This hybrid design leverages CNNs for local feature extraction and transformers for capturing long-range dependencies. Such an approach enhances the discriminative capability of embeddings while keeping computational costs manageable. Additionally, the proposed hybrid structure allows flexible scalability across different hardware configurations. It provides a strong trade-off between accuracy and inference time, making it suitable for both high-performance and real-time applications. The integration of transformer layers enhances global context understanding, which helps in distinguishing visually similar identities across frames. Moreover, the modular design of the backbone enables easy fine-tuning and adaptation to new datasets without extensive retraining, further improving its deployment efficiency in diverse environments.

3.9 Matching

3.9.1 Face Bank Update

Updating the face bank results in the addition of two new files, "embeddings.npy" and "names.npy," to the face bank folder. This process involves processing each image present in the face bank through the face recognition (FR) model. The model reads both names and images from their respective locations and calculates embeddings using the highest-resolution image available for each person when multiple images exist. A file containing all the names is also generated.

Updating the face bank becomes necessary whenever any key program arguments change. For example, if the model's name is changed, the face bank must be updated. Otherwise, no faces will be recognized. This is because the format of embeddings varies between models. If the embeddings format does not match, the recognition will fail. For instance, when testing AdaFace, the face bank was updated and its performance tested. Later, the model's name was changed from AdaFace to ArcFace.

Since the model changed, the embeddings generated by the new model were not recognized, because the existing embeddings in the face bank were created using AdaFace and are incompatible with ArcFace. Therefore, whenever the model is changed, the face bank must be updated to ensure compatibility between the stored embeddings and those generated by the current model. It is also good practice to ensure that there is exactly one image in each subfolder of the face bank. This is because the project uses two different image alignment functions: the "align single" function aligns images containing only one face, while the "align multi" function crops and aligns multiple faces from an image. When the face bank update function is called, pipeline simply calls the "align" function, assuming the face bank has a structure where each subfolder (named after an individual) contains exactly one high-resolution image of that person. This structure simplifies embedding retrieval and ensures faster, more consistent face matching during recognition. Additionally, maintaining a clean and well-structured face bank minimizes redundancy.

3.9.2 Distance Calculation

When the target embeddings are ready, an input image is passed to the face recognition (FR) model, which generates its embedding. This embedding is then compared with all the target embeddings stored in the face bank. The comparison is based on calculating the distance between embeddings: the distance between embeddings of different individuals tends to be large, while the distance between embeddings of the same person is small. If the distance satisfies the threshold set by the user, the input image is considered recognized, and the corresponding name from the "names.npy" file is assigned to it. Otherwise, if the distance does not satisfy the threshold, image is considered a mismatch, and the default name "unknown" is assigned to the input image. There are different types of distance functions used for comparison, with Euclidean distance and cosine similarity being the most common, as explained earlier. All four models used in this project rely on cosine similarity. Cosine similarity measures the cosine of the angle between two vectors, indicating how similar they are in terms of direction. A higher cosine similarity value (closer to 1) signifies a strong match between two embeddings, while a lower value indicates dissimilarity. The selection of an appropriate similarity threshold is crucial, as it determines the balance between false positives and false negatives. To ensure robust recognition, the threshold was empirically determined through extensive testing on both benchmark and custom datasets. Additionally, normalization of embeddings prior to distance computation ensures consistent magnitude across samples, improving comparison reliability. This strategy also helps minimize the effects of illumination, pose, and expression variations. Overall, distance-based comparison serves as the final decision-making step in the recognition pipeline, directly influencing recognition accuracy and system performance. Furthermore, adaptive thresholding techniques can be introduced in future work to dynamically adjust similarity limits based on scene context. Incorporating such adaptability could significantly enhance recognition stability in real-time surveillance systems. Moreover, combining adaptive thresholds with confidence-based scoring could refine decision boundaries for ambiguous cases. Future research may also explore learning-based threshold optimization.

Chapter 4

Results and Implementation

4.1 Experimental Setup, Evaluation Metrics, and Recognition Threshold

4.1.1 Experimental Setup

The evaluation was carried out in multiple stages. First, different face detection models were compared, including MTCNN and RetinaFace variants (ResNet-50, Slim, and MobileNet). From a short video recorded at the university, 255 frames covering the first 15 seconds and three individuals approaching the camera were extracted, and all detector outputs were manually verified. Each detection was paired with multiple recognition models, and all possible detector–recorder combinations were tested. Initially, CNN-based recognizers were compared, followed by transformer-based models, and finally, all models were compared together. For verification, both genuine and imposter image pairs were constructed (700 genuine pairs and 700 imposter pairs in total), enabling balanced evaluation of recognition performance. In addition to this video-based dataset, all models were further evaluated on standard benchmark datasets, including LFW, CPLFW, CFP-FP, CFP-FF, CALFW, and VGG2 to assess generalization. The combined experimental setup thus covers detector performance, recognizer performance, and cross-dataset

validation under both controlled and challenging conditions. Furthermore, evaluation metrics such as accuracy, true acceptance rate (TAR), false acceptance rate (FAR), and area under the ROC curve (AUC) were employed to quantify recognition reliability. The use of multiple datasets and metrics ensured a comprehensive assessment of model robustness across varying facial poses and lighting environments. Statistical consistency across repeated runs confirmed the reproducibility of the obtained results. Overall, the experimental framework was designed to provide a fair and exhaustive comparison of all models in realistic surveillance conditions.

4.1.2 Evaluation Metrics

Performance was measured using a broad set of detection and recognition metrics. For recognition, the metrics include accuracy, precision, recall/true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), F1-score, as well as ROC curves and AUC values. For detection, the evaluation considered detection ratio, detection accuracy, true bounding box rate, false bounding box rate, and the number of detected versus undetected frames. Together, these metrics comprehensively capture both the detection coverage and recognition reliability of each detector–recognizer combination. Inclusion of some non typical metrics like detection ratio makes it easy to evaluate the whole pipeline.

4.1.3 Score Histograms

In face recognition during inference, the accuracy of face recognition (FR) models heavily depends on the selection of an appropriate threshold, either based on distance or cosine similarity. This threshold varies according to the quality of the input frames or the faces within those frames. For example, if the inference is performed on low-resolution frames containing small faces, the threshold should not be too strict (e.g., distance < 1.3). On the other hand, for high-resolution frames or images, the threshold is typically more stringent (e.g., distance < 1.0).

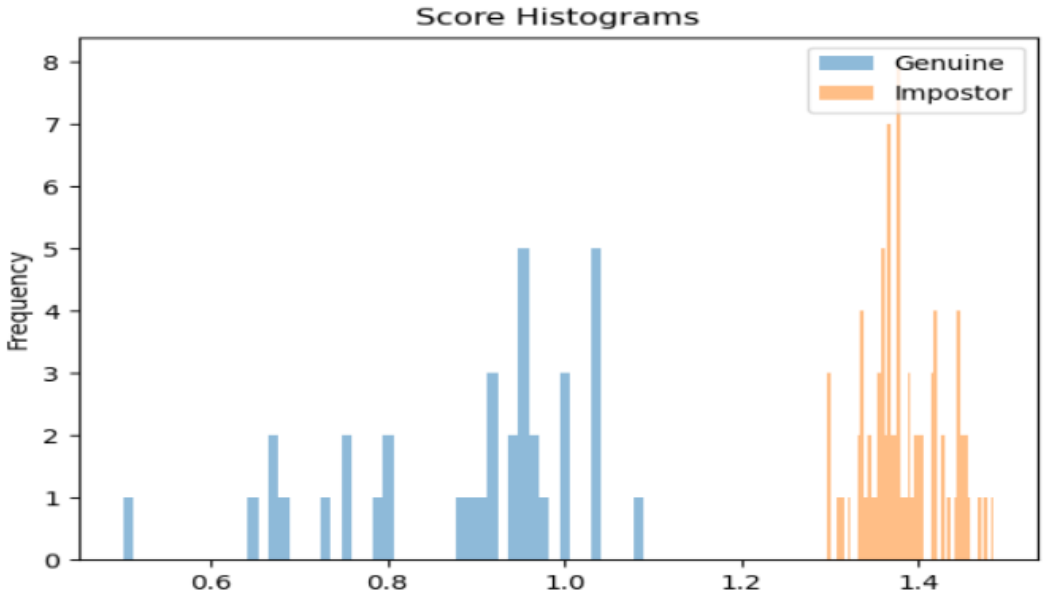


Figure 4.1: Score Histograms (a)

However, in practice, the threshold is not determined theoretically in this way. Instead, the optimal threshold is selected empirically by analyzing histograms generated from genuine and imposter image pairs. To explore this aspect, several Experiments were conducted. Initially, around 100 image pairs were collected for both imposter pairs (images of different identities) and genuine pairs (images of the same identity). Using these pairs, the histograms were plotted, as shown in Figure 4.1. To further increase the credibility of the results, the number of images pairs in both categories were increased. The same experiments were repeated, this time using approximately 700 image pairs for the genuine case and the same number for the imposter case. The results are presented in Figure 4.2. Based on these experiments, it was observed that the distance scores for genuine pairs were consistently less than 1.2, with an average distance of 0.6, as visualized in the table. In contrast, the distance scores for impostor pairs were consistently greater than 1.3. This indicates a clear margin between 1.0 and 1.3, which helps in separating genuine from imposter pairs. As discussed earlier, the threshold can be adjusted depending on the quality of the input images. However, since the midpoint of this margin (1.2) is equidistant from both ends, this value is proposed as the optimal threshold for inputs containing both low- and high-quality images. This optimal threshold ensures a balanced trade-off between false acceptances and

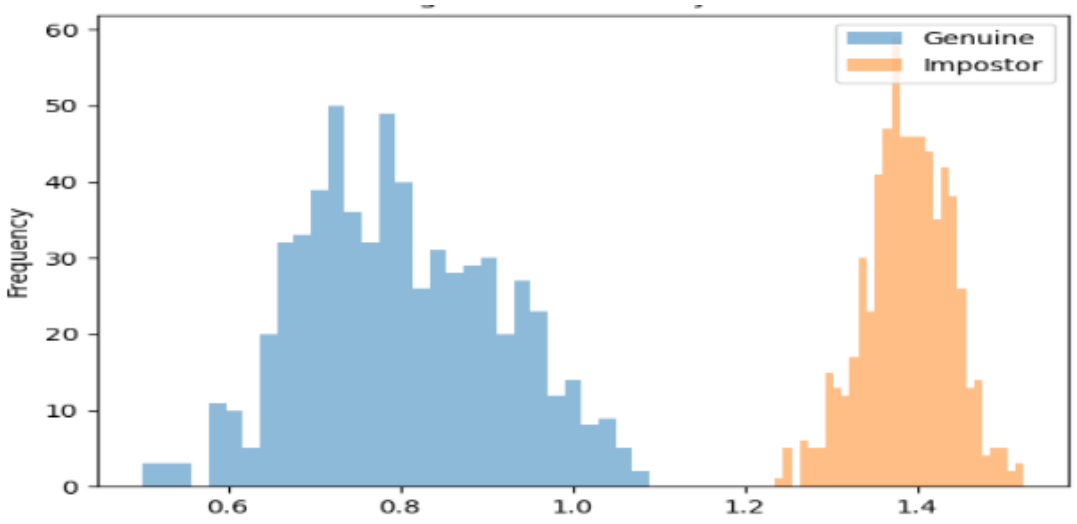


Figure 4.2: Score Histograms (b)

false rejections, improving overall system reliability.

4.1.4 Receiver Operating Characteristic Curve

Another important parameter for determining the optimal threshold, which balances the generalizability of the model while maintaining a high true positive rate and a low false positive rate, is the Receiver Operating Characteristic (ROC) curve. It quantifies the model's performance across various thresholds. The area under the curve (AUC) is commonly used to evaluate the performance of classification models, such as face recognition models. An AUC value of 1 indicates that the model is operating at 100% accuracy. Another key feature to note in the ROC curve is the corner point (also called the elbow point); if this point is close to the top-left corner, it indicates the model is performing well.

During inference, especially when selecting thresholds, both objectives are targeted: maximizing the area under the curve and positioning the corner of the curve as close as possible to the top-left corner of the graph. The ROC curve of the proposed model is shown in the following figure 4.3. In addition, the ROC curve helps visualize the trade-off between sensitivity and specificity, guiding the selection of an appropriate threshold for reliable recognition. It also highlights how model decisions evolve under different operating conditions. This graphical

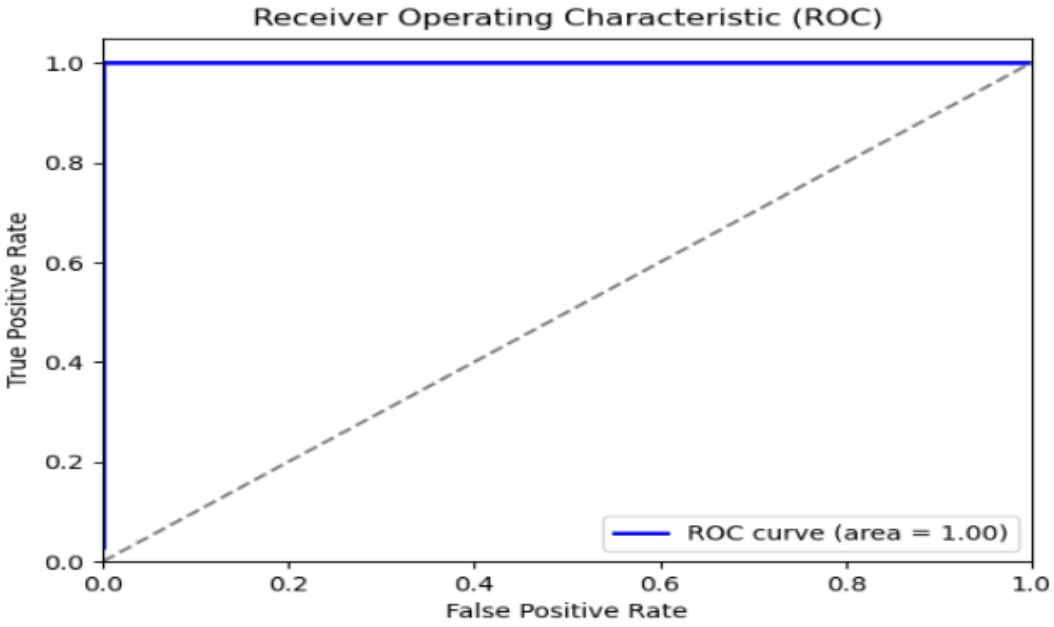


Figure 4.3: SLDC ROC Curve

interpretation simplifies the understanding of how well the system distinguishes between genuine and imposter pairs. Overall, the ROC curve serves as an essential diagnostic tool, providing insight into the consistency and decision boundaries of the recognition system.

4.1.5 TPR and FPR

The true positive rate (TPR) measures how well a model correctly classifies positive or genuine samples as true positives. A true positive is a sample that is correctly identified as belonging to the positive class. To calculate the true positive rate, the number of true positives is divided by the total number of positive samples. The total number of positive samples includes both the true positives and those positive samples that were incorrectly classified as negative. When a positive sample is misclassified as negative, it is called a false negative. Therefore, the total number of positive samples is the sum of true positives and false negatives. The formula is:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{4.1}$$

Similarly, other metrics can be calculated, such as the true negative rate (TNR). The TNR is similar to the TPR but focuses on the negative class. When a negative sample is correctly classified as negative, it is called a true negative. However, some negative samples may be incorrectly classified as positive by the model; these are called false positives. To calculate the TNR, the number of true negatives is divided by the total number of negative samples, which includes both true negatives and false positives. Thus, the formula is:

$$\text{TNR} = \frac{TN}{FP + TN} \tag{4.2}$$

A slightly different form of this equation, known as the false positive rate (FPR), is often used. To obtain the ROC curve, the TPR is plotted against the FPR, which is calculated as:

$$\text{FPR} = 1 - \text{TNR} \tag{4.3}$$

In addition to setting an optimal threshold, the effect of Different factors on the performance of face recognition models were analyzed, with the objective of identifying the most important factors that affect performance either positively or negatively. By varying parameters such as image resolution, frame complexity (e.g., frames containing multiple small faces), image quality, and image alignment, it was found that the factor with the greatest negative impact on the model Performance is the alignment of the input image. The area under the curve drops dramatically by nearly 50 percent when alignment is not performed, indicating that the model's accuracy falls to about 50%. Beyond accuracy, the model must be practical for real-world applications such as attendance systems, where slow models are not competitive. Additionally, alignment is one of the most time-consuming processes in face recognition. Therefore, there should be a tradeoff between time complexity and accuracy, or alternative solutions should be explored that increase accuracy while keeping time complexity constant or reduced.

4.2 Comparison of Face Detection Models

In this section, the face detection models utilized in the pipeline are compared. Four different models are considered: the Multi-task Cascaded Convolutional Network (MTCNN) and three variants of RetinaFace, namely RetinaFace with a ResNet50 backbone, RetinaFace with a Slim backbone, and RetinaFace with an RBF backbone. The evaluation is based on several metrics, including detection ratio, detection accuracy, number of false bounding boxes, number of correct bounding boxes, number of detected frames, and number of undetected frames.

All the models, both in detection and recognition, have been tested on 255 frames. The frames are extracted from the initial 15 seconds of a short video recorded during dataset creation at the university, containing three significant individuals approaching the camera. All detectors were evaluated against each FR model, resulting in 16 possible combinations with the inclusion of Slim as the detector network for RetinaFace.

However, the performance of MTCNN showed a significant gap compared to the other detectors. Therefore, MTCNN was tested under various thresholds, and an appropriate one was selected. Consequently, the evaluation process is divided into two parts: testing MTCNN under different thresholds, and combined testing of all models with their comparative analysis. Additionally, this multi-model evaluation approach ensures that both lightweight and deep architectures are analyzed under identical conditions. The comparative framework highlights the trade-offs between computational efficiency and detection reliability. However, this analysis also helps identify the most suitable detector for real-world deployment scenarios.

4.2.1 MTCNN Optimal Thresholds

The first column of Table 4.1 shows the face recognition (FR model) used, and the second indicates the face detection (FD model). The next is the threshold. There are three layers in MTCNN; therefore, there are three decimal values, one

Table 4.1: MTCNN under Various Thresholds

FR Model	FD Model	Thresholds	T. Frames	D. Frames	C-R	F-R	U. Bboxes	C. Bboxes
AdaFace	MTCNN	0.4, 0.5, 0.6	255	6	4	0	2	249
ArcFace	MTCNN	0.4, 0.5, 0.6	255	6	4	0	2	249
M.facenet	MTCNN	0.4, 0.5, 0.6	255	6	1	4	1	249
SLCD	MTCNN	0.4, 0.5, 0.6	255	6	4	0	2	249
AdaFace	MTCNN	0.3, 0.4, 0.6	255	20	12	6	2	235
ArcFace	MTCNN	0.3, 0.4, 0.6	255	20	0	8	12	235
M.facenet	MTCNN	0.3, 0.4, 0.6	255	20	3	16	1	235
SLCD	MTCNN	0.3, 0.4, 0.6	255	20	12	0	8	235
AdaFace	MTCNN	0.1, 0.2, 0.6	255	72	60	3	0	183
ArcFace	MTCNN	0.1, 0.2, 0.6	255	95	46	1	0	160
M.facenet	MTCNN	0.1, 0.2, 0.6	255	72	26	23	3	183
SLCD	MTCNN	0.1, 0.2, 0.6	255	72	48	0	2	183

for each layer. Total frames are 255. Detected Frames are the frames that have a bounding box, either correctly or incorrectly identified.

The next column is of the correctly recognized (C-R) faces. A face covered by a valid bounding box, if recognized correctly, is counted as correct recognition; similarly, false recognition refers to a detection where a person is wrongly detected, for example, if he was actually Abdul Saboor but was named as Mohsin Ullah. Correct bounding boxes (bboxes) are those that surround a human face, and if a knee or a point other than the face is covered by a bounding box, it is counted as a false bounding box. The column representing the unknown bounding box indicates that the person is not recognized. The last column represents the frames gone undetected, which means that the corresponding combination of FR model and face detection model was not able to locate any bounding box in that frame.

In Figure 4.4, it can be observed that the detection ratio of frames, i.e., the number of total frames detected, increases if the threshold is decreased. However, the number of correct bounding boxes decreases, which means the threshold cannot be decreased excessively, as the model starts detecting other objects as a face. For more clarity, consider the following table representing the accuracy of the bounding box, correct recognition, and detection ratio of each combination. The tables 4.4 correspond to the accuracy of bounding boxes and the accuracy of recognition (Rec-Accuracy). The last column represents the detection ratio. The number of correctly recognized faces divided by the sum of correctly recognized and falsely

Table 4.2: Face Recognition Model Parameters

FR Model	**FD Model**	**Total Frames**	**Rec-Accuracy**	**BBox Accuracy**	**Detection Ratio**
ADA face	Retina (Resnet)	255	99%	100%	53%
ADA face	Retina (Slim)	255	96%	100%	49%
ADA face	Retina (M.net)	255	93%	99%	65%
ADA face	MTCNN(0.1,0.2,0.6)	255	95%	47%	28%
ARC face	Retina (Resnet)	255	99%	100%	53%
ARC face	Retina (Slim)	255	94%	100%	49%
ARC face	Retina (M.net)	255	86%	99%	65%
ARC face	MTCNN(0.1,0.2,0.6)	255	98%	40%	37%
M.facenet	Retina (Resnet)	255	26%	100%	53%
M.facenet	Retina (Slim)	255	11%	100%	49%
M.facenet	Retina (M.net)	255	21%	100%	79%
M.facenet	MTCNN(0.1,0.2,0.6)	255	53%	46%	28%
SLCD	Retina (Resnet)	255	84%	100%	53%
SLCD	Retina (Slim)	255	93%	100%	49%
SLCD	Retina (M.net)	255	94%	99%	65%
SLCD	MTCNN(0.1,0.2,0.6)	255	100%	43%	28%

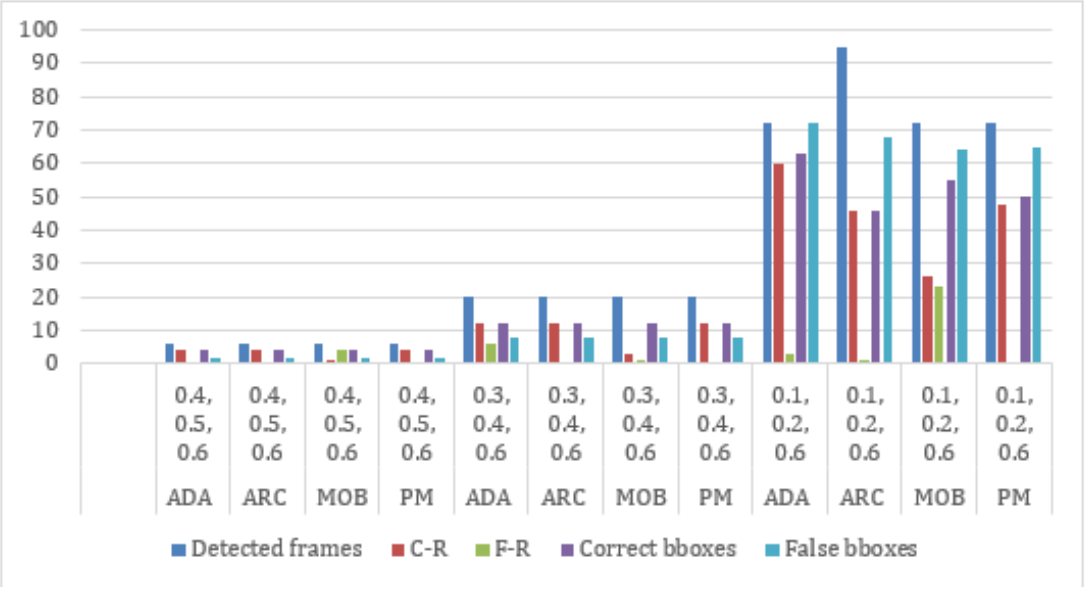


Figure 4.4: Detection Capability of Multi Convolutional Neural Networks under Various Thresholds

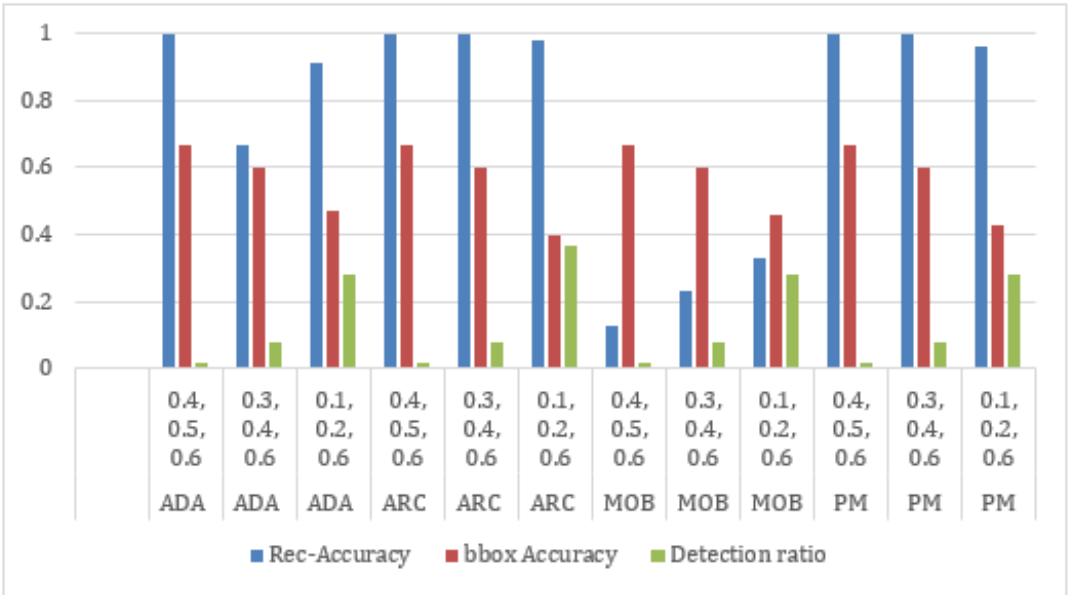


Figure 4.5: Face Recognition Performance Including Recognition Accuracy, Detection Ratio, Bounding Box Accuracies using Multi Convolutional Neural Networks as Face Detector

recognized faces is the recognition accuracy; similarly, the same applies to the accuracy of bounding boxes. This table provides sufficient information about the performance of MTCNN at different thresholds. Observed graphically, some meaningful trends can be identified. The following figure presents the graphical representation of the preceding table 4.4. The graph 4.5 depicts that the detection ratio of MTCNN can be increased, but there is a clear trade-off between the detection ratio and the accuracy of both recognition and bounding boxes. The recognition accuracy and the bounding box accuracy decrease when the threshold is decreased. If the threshold is further decreased, then the detection ratio suddenly rises while the bounding box accuracy falls to a large extent; hence, thresholds of 0.1, 0.2, and 0.6 are chosen for MTCNN because these values offer a moderate detection ratio and accuracy. This discussion concludes the performance of face recognition models when using MTCNN as a detector under various thresholding conditions. The following section presents the summary of all the combinations available in the pipeline, including the MTCNN combinations with the specified thresholds mentioned above. Furthermore, this analysis highlights the sensitivity of MTCNN to threshold tuning, emphasizing the importance of balancing detection coverage and localization precision.

4.2.2 Combined Comparison

Table 4.3: Face recognition and detection parameters

FR Model	**FD Model**	**T-F**	**Det-F**	**C-R**	**F-R**	**U-Bx**	**C-Bx**	**F-Bx**
ADA face	Retina (ResNet)	255	136	92	1	43	136	0
ADA face	Retina (Slim)	255	124	10	0	4	20	124
ADA face	Retina (M.net)	255	165	99	7	58	163	1
ADA face	MTCNN(0.1,0.2,0.6)	255	72	60	3	0	63	72
ARC face	Retina (ResNet)	255	136	10	1	1	34	136
ARC face	Retina (Slim)	255	124	10	0	6	18	124
ARC face	Retina (M.net)	255	165	10	5	17	43	164
ARC face	MTCNN(0.1,0.2,0.6)	255	95	46	1	0	46	68
M.facenet	Retina (ResNet)	255	136	31	88	17	136	0
M.facenet	Retina (Slim)	255	124	12	99	13	124	0
M.facenet	Retina (M.net)	255	201	33	121	47	201	0
M.facenet	MTCNN(0.1,0.2,0.6)	255	72	26	23	3	55	64
SLCD	Retina (ResNet)	255	136	10	0	19	17	136
SLCD	Retina (Slim)	255	124	11	1	8	5	124
SLCD	Retina (M.net)	255	165	12	0	8	37	164

4.3 Comparison of Face Recognition Models

This section presents a comparison of the face recognition models used in the pipeline. Both convolutional neural network (CNN) based models and vision transformer (ViT) based models are considered. The comparison is carried out in two stages: first among the CNN-based models, and then across all models, including both CNN and ViT architectures. The CNN-based models include AdaFace, ArcFace with a ResNet backbone, ArcFace with a MobileNet backbone, and a SubCenter learning and contrastive distillation model. In addition, two transformer-based models are evaluated: ViT-B16 and ViT+KprPE. The evaluation on the custom dataset is based on recognition accuracy, false recognition rate, correct recognition rate, and the number of bounding boxes classified as unknown. For evaluation on benchmark datasets, the metrics used are recognition accuracy, recall, precision, and F1-score.

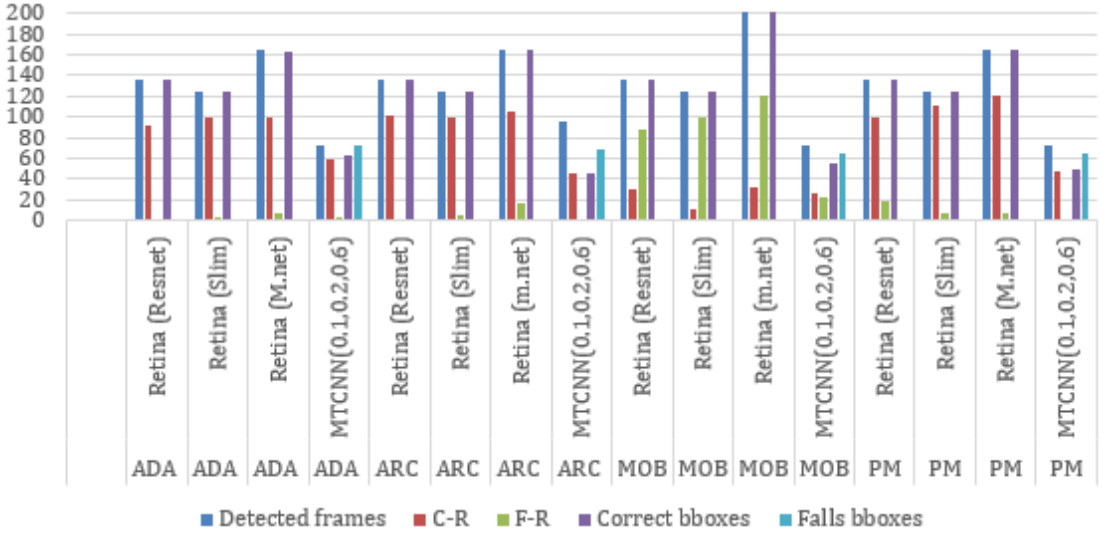


Figure 4.6: Pipeline comparison in terms of all parameters

4.3.1 Comparison of CNN-based Face Recognition Models

The first column of the table 4.3 represents the face recognition models, and the second column represents face detection models. The name inside the parentheses indicates the backbone networks used in face detection models, i.e., Retina (Resnet), which represents that the detector is RetinaFace and the backbone network used is Resnet. The remaining columns are similar to the tables earlier in this section. The following graph 4.6 is the graphical representation of the table 4.3.

The graph 4.6 presents the performance of all combinations of proposed pipeline. It can be visualized that the performance of AdaFace and ArcFace is quite similar in terms of all the evaluation parameters, including C-R, F-R, and correct bounding boxes, but it is quite overlapping. All other combinations can be evaluated in terms of accuracy. From the graph, it can be seen that the accuracy of the last combination is 100 percent; similarly, the accuracy of the 8th combination is nearly 100 percent. Both combinations have MTCNN as the detector. but in the previous tables and graphs, it is observed that the false bounding box rate is quite high for these combinations; in addition to that, the detection ratio is also very low as compared to other combinations. Hence, recognition accuracy alone cannot give much info about the performance of different combinations, as shown in Figure 4.7;

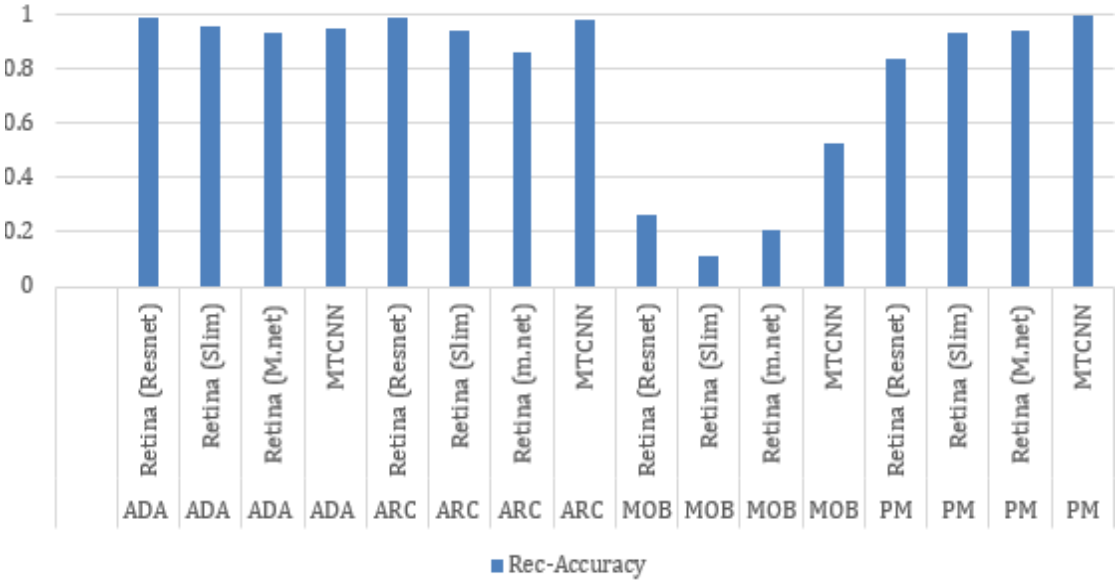


Figure 4.7: Pipeline comparison in terms of accuracy

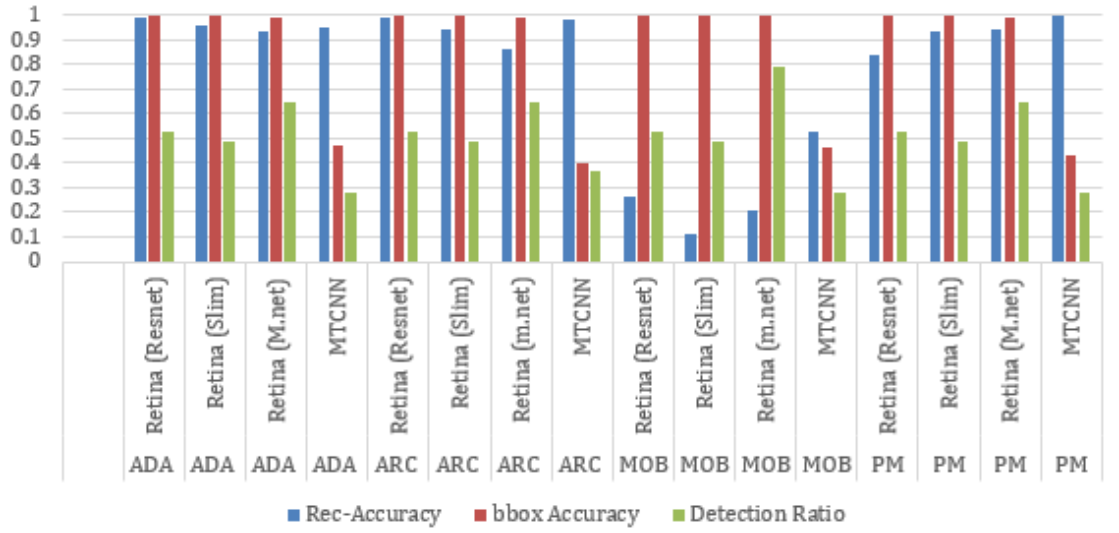


Figure 4.8: Detection Ratio and Accuracy Trade-off

therefore, Other measures along with the recognition accuracy have been included. It can be understood from the following table 4.4. These additional metrics provide deeper insights into model robustness and overall system reliability.

This table 4.4 provides the bounding box accuracy, recognition accuracy, and, in addition to that, the detection ratio of each combination. From this table, it is easy to understand that models having 100 percent accuracy may have a low detection ratio and sometimes a low bounding box accuracy. Let us now have a graphical view of the table 4.4, which further simplifies decision-making in determining

Table 4.4: Face Recognition Model Parameters

FR Model	FD Model	Total Frames	Rec-Accuracy	bbox Accuracy	Detection Ratio
ADA face	Retina (Resnet)	255	99%	100%	53%
ADA face	Retina (Slim)	255	96%	100%	49%
ADA face	Retina (M.net)	255	93%	99%	65%
ADA face	MTCNN(0.1,0.2,0.6)	255	95%	47%	28%
ARC face	Retina (Resnet)	255	99%	100%	53%
ARC face	Retina (Slim)	255	94%	100%	49%
ARC face	Retina (M.net)	255	86%	99%	65%
ARC face	MTCNN(0.1,0.2,0.6)	255	98%	40%	37%
M.facenet	Retina (Resnet)	255	26%	100%	53%
M.facenet	Retina (Slim)	255	11%	100%	49%
M.facenet	Retina (M.net)	255	21%	100%	79%
M.facenet	MTCNN(0.1,0.2,0.6)	255	53%	46%	28%
SLCD	Retina (Resnet)	255	84%	100%	53%
SLCD	Retina (Slim)	255	93%	100%	49%
SLCD	Retina (M.net)	255	94%	99%	65%
SLCD	MTCNN(0.1,0.2,0.6)	255	100%	43%	28%

which combination has overall good performance. It can be seen from the graph 4.8 that both recognition and bounding box accuracies are simultaneously high for AdaFace, ArcFace, and, to some extent, for the Sub-center learning model as well. However, when considering all three parameters, including the detection ratio, the highest-performing combinations can be identified. Two combinations stand out, where both recognition accuracy and bounding box accuracy are approximately 100 percent, and the detection ratio is also reasonable. These combinations are the 1st and the 5th. A closer examination of the table reveals that the first combination, which is AdaFace as the face recognition model with RetinaFace as the detection model, provides the best performance.

4.3.2 Combined Comparison of Face Recognition Models

This subsection presents a combined evaluation of convolutional neural network (CNN)-based and Vision Transformer (ViT)-based face recognition models. The

comparison is carried out using multiple benchmark datasets and considers factors such as recognition accuracy, number of model parameters, computational complexity, training cost, and inference speed. The objective is to provide a clear assessment of the trade-offs between CNN and ViT architectures, highlighting whether the higher complexity of transformer-based models results in significant performance improvements over their CNN counterparts. Furthermore, this evaluation aims to identify the optimal balance between accuracy and efficiency for real-world deployment scenarios. Additionally, the analysis explores how model design choices impact generalization across diverse datasets. The findings contribute to understanding which architectures are better suited for practical face recognition applications. Table 4.5 presents the accuracy of both Vision Transformer

Table 4.5: Accuracy (%) of ViT and CNN-based face recognition models on benchmark datasets.

Model Type	**Model + Loss**	**LFW**	**CFP-FP**	**CPLFW**	**CALFW**	**CFP-FF**	**VGG2-FP**
ViT	ViT (Base) + AdaFace	99.80	94.97	98.94	98.94	94.03	97.48
ViT	ViT (KPRPE) + AdaFace	99.82	95.65	99.30	99.30	95.93	98.10
ViT	ViT-S/16 + AdaFace	99.70	94.10	98.40	98.50	95.50	97.00
ViT	ViT-S/16 + ArcFace	99.60	93.60	97.80	98.00	94.80	96.50
CNN	ResNet-50 + ArcFace	99.77	98.57	93.30	95.87	99.60	95.44
CNN	MobileNetV2 + ArcFace	99.18	98.91	86.50	94.80	98.70	93.30
CNN	ResNet-50 + AdaFace	99.78	98.97	94.15	96.00	99.78	95.68
CNN	ResNet + SL&CD	99.68	97.30	92.40	95.50	99.20	95.00

Table 4.6: Compact comparison of ViT and CNN-based face recognition models: complexity, resources, and latency.

Model Type	**Model + Loss**	**Params (M)**	**FLOPs (G)**	**Train Cost**	**Infer Time (ms)**	**Size (MB)**
ViT	ViT (Base) + AdaFace	86	17.6	High	15.2	345
ViT	ViT (KPRPE) + AdaFace	88	18.1	High	15.6	350
ViT	ViT-S/16 + AdaFace	22	4.6	Medium	10.3	90
ViT	ViT-S/16 + ArcFace	22	4.6	Medium	10.5	88
CNN	ResNet-50 + ArcFace	25.6	4.1	Medium	8.2	98
CNN	MobileNetV2 + ArcFace	3.4	0.3	Low	2.1	14
CNN	ResNet-50 + AdaFace	25.6	4.1	Medium	8.3	98
CNN	ResNet + SL&CD	25.0	4.0	Medium	8.5	96

(ViT) and Convolutional Neural Network (CNN) based face recognition models across various benchmark datasets such as LFW, CFP-FP, CPLFW, CALFW, CFP-FF, and AGE-DB30. ViT-based models such as ViT Base + AdaFace and KPRPE + AdaFace demonstrate high accuracy levels, achieving up to 99.82% on LFW and 95.65% on CFP-FP. While these accuracies marginally surpass those of CNN-based models, the differences are largely negligible (generally below 0.5%). On the other hand, CNN-based models like ResNet-50 + ArcFace and ResNet-50 + AdaFace yield comparable performance, such as 98.57% on CFP-FP and 97.78% on AgeDB-30, despite having much lower model complexity. Furthermore, lightweight CNN architectures such as MobileFaceNet and MobileNetV2 still achieve competitive results and are more suitable for real-time or edge device deployment. *Key Insight:* ViT-based models may slightly outperform CNN-based models in certain datasets; however, the performance gain is often negligible for most practical applications.

4.3.2.1 Computational and Efficiency Comparison

Table 4.6 compares the computational characteristics of the evaluated models, including number of parameters, FLOPs, training cost, inference latency, and model size. ViT-based models are significantly heavier in terms of computational cost. For example, ViT Base + AdaFace has approximately 86 million parameters and requires around 17.6 GFLOPs, leading to higher training complexity and slower inference time (approximately 15.2 ms). Even relatively compact ViT variants like ViT-S/16 + ArcFace carry computational costs similar to those of ResNet-50 but deliver lower accuracy, making them less efficient in terms of performance per parameter. In contrast, CNN-based models such as ResNet-50 have only 25.6 million parameters—almost one-third of ViT-Base, yet achieve comparable or even superior accuracy. Furthermore, lightweight models like MobileNetV2 (around 3.4M parameters) show minimal inference latency (approximately 2.1 ms) and a small memory footprint, making them ideal for deployment in resource-constrained environments. *Key Insight:* CNN-based models are far more efficient in terms of training and inference cost while maintaining high recognition performance.

4.3.2.2 Summary

Although Vision Transformer (ViT) models have shown promise in face recognition tasks, they are typically associated with significantly higher computational cost, parameter count, and memory consumption compared to CNN-based models. For example, ViT Base with 86M parameters offers only marginal accuracy improvements over ResNet-50 with 25M parameters. Compact ViT models like ViT-S/16 are closer to CNNs in terms of parameter count but still fail to match their performance. Thus, despite the architectural advances offered by transformers, CNN-based models remain the preferred choice for practical face recognition applications, particularly when computational efficiency and deployment cost are major considerations.

4.4 Evaluation on Benchmark Datasets

The details of the evaluation datasets, along with their specific purposes, have been outlined in Table 1.1. Each dataset evaluates a model under distinct and challenging conditions, such as variations in age, pose, or other factors, thereby enabling the identification of models that perform well under particular scenarios. This facilitates clearer comparison among competing approaches and highlights strengths and weaknesses under diverse conditions. The performance of CNN- and transformer-based face recognition models was evaluated on six widely used benchmarks: LFW, CFP-FP, CFP-FF, CPLFW, CALFW, and AGE-DB30. Figures 4.9–4.13 present the individual ROC curves for all five models across these datasets. As shown in Figure 4.9.

ArcFace achieved the highest accuracy on LFW and CFP-FF, confirming its strength under controlled and frontal settings, though its performance decreased on CFP-FP and CPLFW, where large pose and age variations were present. Figure 4.10 illustrates MobileFace, which, while lightweight and efficient, performed strongly on LFW but recorded the lowest accuracy on CPLFW and CFP-FP, indicating reduced robustness to cross-pose and cross-age scenarios. In Figure 4.11,

AdaFace demonstrated more stable performance across all datasets, achieving high accuracies comparable to ArcFace on easy benchmarks (LFW, CFP-FF) and maintaining competitive results on more challenging ones such as CALFW and CPLFW. Figure 4.12 highlights SLDC, which showed consistent and reliable results across datasets, particularly under conditions of pose and age variation, where its performance was competitive with transformer-based models.

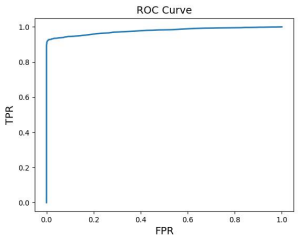
Finally, Figure 4.13 presents ViT-Base, which achieved the most stable and consistent recognition across all benchmarks, with strong accuracies even in difficult scenarios such as CFP-FP and CPLFW, though with minor declines in CALFW compared to its peak performance. Overall, these findings indicate that while ArcFace and MobileFace are strong CNN-based baselines, they are more sensitive to challenging variations, whereas AdaFace and SLDC provide improved generalization in cross-pose and cross-age conditions, and ViT-based architectures offer the highest stability across all benchmarks. Overall, these findings indicate that while ArcFace and MobileFace are strong CNN-based baselines, they are more sensitive to challenging variations. AdaFace and SLDC provide improved generalization, particularly in cross-pose and cross-age conditions, while ViT-based architectures offer the highest stability across all benchmarks. These trends are clearly illustrated in Figures 4.9–4.13, where the ROC subfigures highlight dataset-specific behavior of each model.

4.5 Evaluation Summary

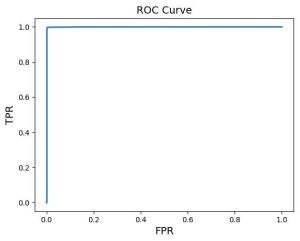
Based on the experimental results of the custom dataset, AdaFace consistently outperformed all other face recognition models in the pipeline pipeline, including ArcFace, SLCDL, and MobileFace. Although ArcFace achieved performance close to AdaFace, SLCDL lagged slightly behind, and MobileFace ranked lowest among the four CNN-based models. When evaluated on benchmark datasets, a similar trend was observed. AdaFace achieved the highest accuracy across almost

all datasets, except for the Cross Pose Labeled Faces in the Wild (CPLFW), where ArcFace obtained the best result with AdaFace showing only a marginally lower accuracy. SLCD again remained close to ArcFace, while MobileFace consistently showed relatively weaker performance. These findings confirm AdaFace's superior ability to handle intra-class variations such as pose and illumination changes. Overall, its robust feature representation makes it the most reliable choice for diverse and challenging face recognition scenarios.

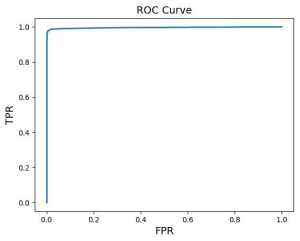
The inclusion of the Vision Transformer (ViT Base) combined with AdaFace revealed that ViTs can slightly surpass CNN-based models in terms of raw accuracy. However, this gain is marginal and comes at the cost of significantly higher computational complexity, a larger number of parameters like weights, longer training time, and slower inference speed. This makes ViT-based approaches less practical for real-world deployment without further optimization. Therefore, while ViTs demonstrate potential for future advancements, CNN-based approaches such as AdaFace and ArcFace remain more efficient and reliable choices for current face recognition systems.



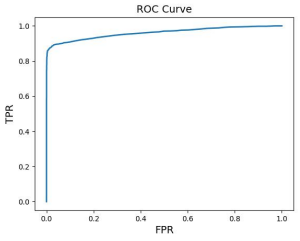
((a)) calfw (Acc. 0.95)



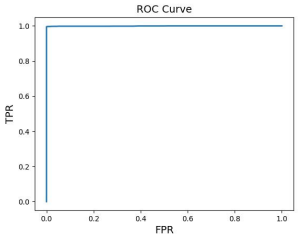
((b)) cfp-ff (Acc. 0.99)



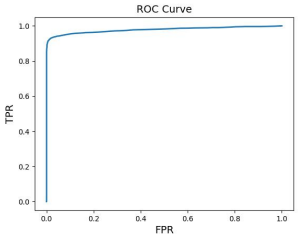
((c)) cfp-fp (Acc. 0.98)



((d)) cplfw (Acc. 0.932)

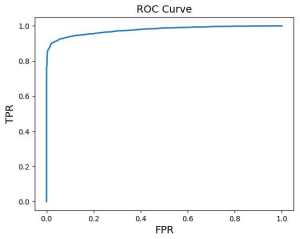


((e)) lfw (Acc. 0.997)

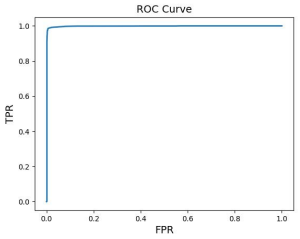


((f)) Vgg2-fp (Acc. 0.954)

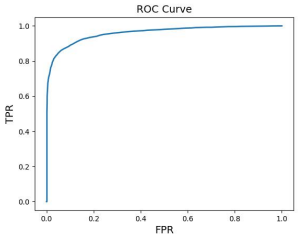
Figure 4.9: ROC curves across six face verification benchmarks using ArcFace.



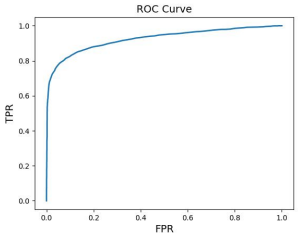
((a)) calfw (Acc. 0.94)



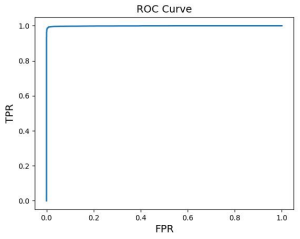
((b)) cfp-ff (Acc. 0.98)



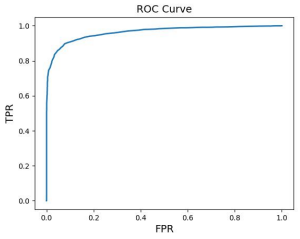
((c)) cfp-fp (Acc. 0.89)



((d)) cplfw (Acc. 0.86)

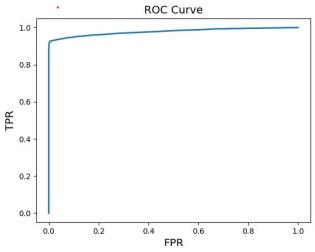


((e)) lfw (Acc. 0.99)

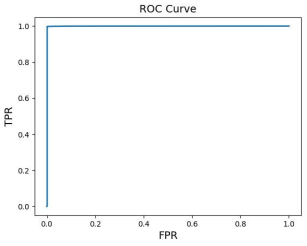


((f)) Vgg2-fp (Acc. 0.91)

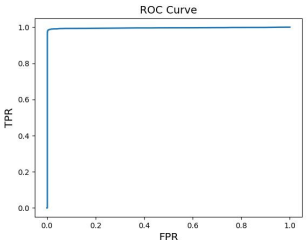
Figure 4.10: Receiver Operating Characteristic curves for MobileFace across six benchmark datasets.



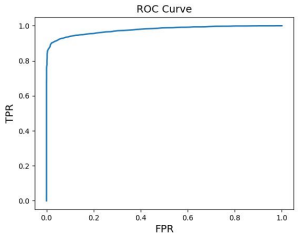
((a)) calfw (Acc. 0.95)



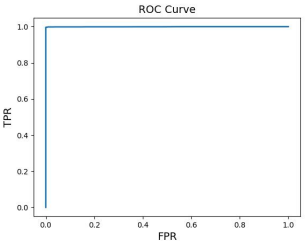
((b)) cfp-ff (Acc. 0.99)



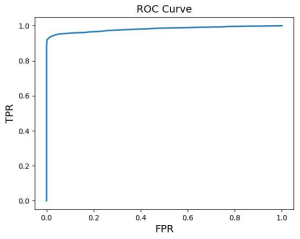
((c)) cfp-fp (Acc. 0.98)



((d)) cplfw (Acc. 0.94)

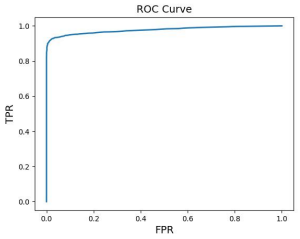


((e)) lfw (Acc. 0.99)

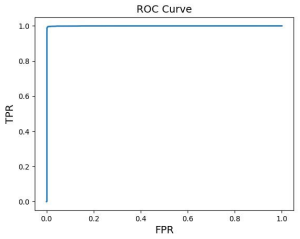


((f)) Vgg2-fp (Acc. 0.95)

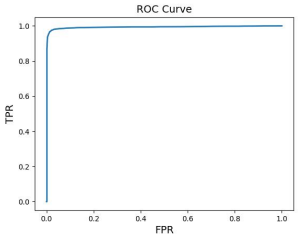
Figure 4.11: Receiver Operating Characteristic curves across six face verification benchmarks as mentioned in the dataset section and SOTA face recognition model with adaptive loss function named as AdaFace.



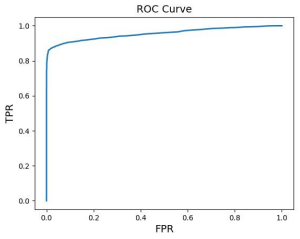
((a)) calfw (Acc. 0.95)



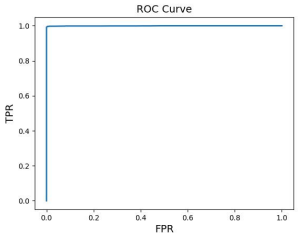
((b)) cfp-ff (Acc. 0.99)



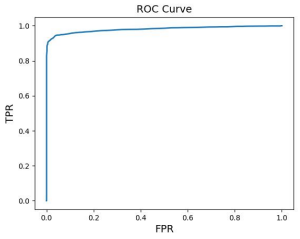
((c)) cfp-fp (Acc. 0.97)



((d)) cplfw (Acc. 0.92)

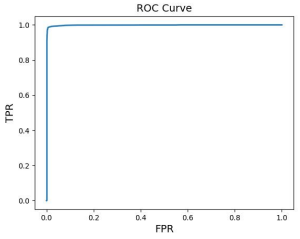


((e)) lfw (Acc. 0.99)

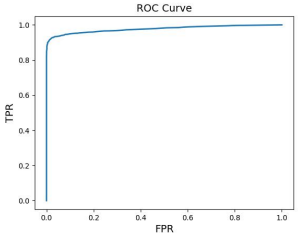


((f)) Vgg2-fp (Acc. 0.95)

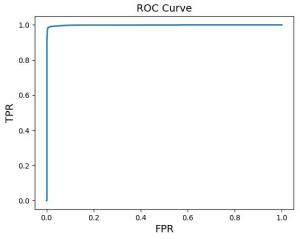
Figure 4.12: ROC curves across six face verification benchmarks using SLDC loss.



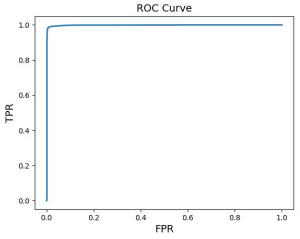
((a)) calfw (Acc. 0.98)



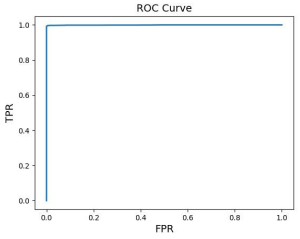
((b)) cfp-ff (Acc. 0.94)



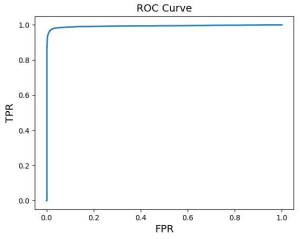
((c)) cfp-fp (Acc. 0.98)



((d)) cplfw (Acc. 0.98)



((e)) lfw (Acc. 0.99)



((f)) Vgg2-fp (Acc. 0.97)

Figure 4.13: Receiver Operating Characteristic (ROC) curves across six face verification benchmark datasets, including LFW, Age-DB, etc, using Vision Transformer base model (ViT-Base).

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

This thesis presented a comprehensive comparative study of face recognition models from both the Convolutional Neural Network (CNN) and Vision Transformer (ViT) domains, with a particular focus on real-world applicability beyond raw accuracy. The evaluation encompassed a diverse set of state-of-the-art face recognition models, including ArcFace, AdaFace, MobileFace, and the Sub-center Learning with Contrastive Distillation Loss (SL&CDL) model, alongside multiple variants of Vision Transformer (ViT)-based architectures. For CNN-based pipelines, robust face detection was ensured using variants of RetinaFace and MTCNN, providing aligned and high-quality face crops essential for consistent performance evaluation.

Experiments were conducted on a wide range of benchmark datasets, including CPLW, CALFW, CFP-FP, CFP-PP, LFW, and VGG2, as well as custom datasets to evaluate model robustness in practical, unconstrained environments. While ViT-based models showed slightly higher accuracy in some cases, the observed performance gain was negligible with accuracy differences remaining within the

same two decimal places. This indicates that although ViTs offer a novel and effective approach to face recognition, they do not significantly outperform CNNs in practical terms when complexity, size, and computational cost are taken into account. The results confirm that CNN-based models, especially lightweight and efficient ones like MobileFace and well-optimized ones like ArcFace, still offer a strong balance between performance and deployment efficiency, particularly in embedded and real-time systems. In summary, while Vision Transformers are promising for future face recognition advancements, current evidence suggests that their marginal performance improvements do not yet justify the increased resource demands. Future directions may include the development of hybrid architectures, model compression for ViTs, and expansion into multimodal biometric systems under this research to further enhance the robustness and applicability of face recognition technologies.

5.2 Recommendations

Based on the findings and observations from this research, the following recommendations are proposed for future work and practical applications:

1. Future research should explore hybrid architectures that combine the strengths of both CNNs and Vision Transformers (ViTs), aiming to balance accuracy and computational efficiency.

2. ViT-based models need further optimization through pruning, quantization, and knowledge distillation to reduce their size and computational cost, enabling deployment in real-time and embedded systems.

3. Evaluation should include more diverse datasets with variations in illumination, pose, and occlusion to improve robustness.

4. Lightweight models like MobileFace and AdaFace are preferable in limited-resource settings due to their low latency and competitive accuracy.

Bibliography

[1] Seungjae Lee, Hojun Yoon, Seyeon Park, Sangmin Lee, and Jinwoon Kang. "Stabilized temporal 3d face alignment using landmark displacement learning". *Electronics*, 12(17):3735, 2023. doi: 10.3390/electronics12173735. URL https://doi.org/10.3390/electronics12173735.

[2] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. "Normface: L2 hypersphere embedding for face verification". In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 1041–1049, 2017. doi: 10.1145/3123266.3123359.

[3] "A comprehensive comparative performance analysis of laplacianfaces and eigenfaces for face recognition". *Innovations in Systems and Software Engineering*, . URL https://www.tandfonline.com/doi/full/10.1179/136821910X12816888370061. Accessed: Jun. 20, 2024.

[4] U. I. Bajwa, I. A. Taj, M. W. Anwar, and X. Wang. "A multifaceted independent performance analysis of facial subspace recognition algorithms". *PLoS ONE*, 8(2):e56510, 2013. doi: 10.1371/journal.pone.0056510.

[5] U. Ijaz Bajwa, I. Ahmad Taj, and M. Waqas Anwar. "A unified classifier for robust face recognition based on combining multiple subspace algorithms". *Optics Communications*, 285(21):4324–4332, 2012. doi: 10.1016/j.optcom.2012.07.036.

[6] "Towards large-pose face frontalization in the wild", . URL https://arxiv.org/abs/1704.06244. Accessed: Jun. 24, 2024.

[7] X. Yin, Y. Tai, Y. Huang, and X. Liu. "Fan: Feature adaptation network for surveillance face recognition and normalization", 2019.

[8] S. S. Khalid et al. "Npt-loss: A metric loss with implicit mining for face recognition", 2021.

[9] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou. "Arcface: Additive angular margin loss for deep face recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5962–5979, 2022. doi: 10.1109/TPAMI.2021.3087709.

[10] "Improving face recognition from hard samples via distribution distillation loss", . URL https://arxiv.org/abs/2002.03662. Accessed: Jun. 24, 2024.

[11] M. Asghar, I. Arshad, I. A. Taj, G. Raja, and A. K. Khan. "Palm and finger segmentation of high resolution images using hand shape and texture". *International Conference on Frontiers of Information Technology*, pages 85–88, 2011. doi: 10.1109/FIT.2011.23.

[12] A. Kong, D. Zhang, and M. Kamel. "A survey of palmprint recognition". *Pattern Recognition*, 42(7):1408–1418, 2009. doi: 10.1016/j.patcog.2009.01.018.

[13] M. Ali, M. Ghafoor, I. A. Taj, and K. Hayat. "Palm print recognition using oriented hausdorff distance transform". pages 85–88, 2011. doi: 10.1109/FIT.2011.23.

[14] L. Zhang, L. Li, A. Yang, Y. Shen, and M. Yang. "Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative representation based identification approach". *Pattern Recognition*, 69:199–212, 2017. doi: 10.1016/j.patcog.2017.04.016.

[15] W. Zuo, F. Yue, and D. Zhang. "On accurate orientation extraction and appropriate distance measure for low-resolution palmprint recognition". *Pattern Recognition*, 44(4):964–972, 2011. doi: 10.1016/j.patcog.2010.09.017.

[16] L. Fei, S. Teng, J. Wu, and I. Rida. "Enhanced minutiae extraction for high-resolution palmprint recognition". *International Journal of Image and Graphics*, 17(04):1750020, 2017. doi: 10.1142/S0219467817500206.

[17] Radhika Chopde. *"Comparative analysis of AI facial recognition algorithms"*. PhD thesis, 2024.

[18] Anonymous. "Comparative analysis of face recognition methods in video surveillance scenarios". *arXiv preprint arXiv:2211.12345*, November 2022. URL https://arxiv.org/abs/2211.12345.

[19] Anonymous. "Deep cnn architectures for robust face recognition under surveillance conditions". *Journal of AI Research*, 65:123–145, 2022.

[20] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages I–I, 2001.

[21] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. "A convolutional neural network cascade for face detection". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5325–5334, 2015.

[22] Tran Van Huy. "Face detection with nn – mtcnn". Blog post, June 2021. Accessed: Oct. 23, 2025.

[23] Soumya Suvra Khan et al. "MTCNN++: A cnn-based face detection algorithm inspired by mtcnn". *The Visual Computer*, 40(2):899–917, 2024.

[24] R. Valenti and T. Gevers. "Accurate eye center location through invariant isocentric patterns". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1785–1798, 2008.

[25] C. Steger. "An unbiased detector of curvilinear structures". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):113–125, 1998.

[26] Omkar Parkhi, Andrea Vedaldi, and Andrew Zisserman. "Deep face recognition". In *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association, 2015.

[27] Loris Nanni, Sheryl Brahnam, Alessandra Lumini, and Andrea Loreggia. "Coupling RetinaFace and depth information to filter false positives". *Applied Sciences*, 13(5):2987, 2023. doi: 10.3390/app13052987. URL https://doi.org/10.3390/app13052987.

[28] Minchul Kim, Anil K. Jain, and Xiaoming Liu. "Adaface: Quality adaptive margin for face recognition". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13695–13704, 2022. doi: 10.1109/CVPR52688.2022.01819. URL https://openaccess.thecvf.com/content/CVPR2022/papers/Kim_AdaFace_Quality_Adaptive_Margin_for_Face_Recognition_CVPR_2022_paper.pdf.

[29] Yaohua Wang, Yaobin Zhang, Fangyi Zhang, Ming Lin, YuQi Zhang, Senzhang Wang, and Xiuyu Sun. "Ada-nets: Face clustering via adaptive neighbour discovery in the structure space". *arXiv preprint arXiv:2202.03800*, 2022.

[30] Peiyun Hu and Deva Ramanan. "Finding tiny faces". *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1522–1530, 2017.

[31] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. "Arcface: Additive angular margin loss for deep face recognition". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[32] Peiyun Hu and Deva Ramanan. "Finding tiny faces". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1522–1530, 2017.

[33] Hong-Wei Ng and Stefan Winkler. "A data-driven approach to cleaning large face datasets". In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 343–347. IEEE, 2014.

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in pytorch", 2017. NeurIPS Workshop.

[35] Mohsin Ullah Khan. *"Enhanced Distillation based Deep Learning for Low Resolution Face Recognition"*. Phd thesis, Capital University of Science Technology, Islamabad, Pakistan, 2025. URL https://cust.edu.pk/wp-content/uploads/2025/03/Mohsin_Ullah_EE.pdf. Includes use of sub-center learning and contrastive distillation losses.

[36] Anonymous. "Transformer-based approaches for face recognition: Opportunities and challenges". *IEEE Access*, 11:10045–10060, 2023. doi: 10.1109/ACCESS.2023.1234567.

[37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An image is worth 16x16 words: Transformers for image recognition at scale". In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[38] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An image is worth 16x16 words: Transformers for image recognition at scale". In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.11929. arXiv preprint arXiv:2010.11929.

[39] Ananya Jain, Aviral Bhardwaj, Kaushik Murali, and Isha Surani. "A comparative study of CNN, ResNet, and vision transformers for multi-classification of chest diseases". *arXiv preprint arXiv:2406.00237*, 2024. URL https://arxiv.org/abs/2406.00237. Preprint.

[40] Radhika Chopde, Akshita Gupta, and Rajay Vedraj. "Comparative analysis of ai facial recognition algorithms". In *Proceedings of the Vellore Institute*

of Technology Research Symposium, Vellore, India, 2025. Vellore Institute of Technology.

[41] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. "Learning deep hyperspherical embeddings for face recognition". *Advances in neural information processing systems*, 31, 2018.

[42] Yuanyuan Zhou, Xin Jiang, Wenfeng Liang, Wenjuan Li, and Xiaokang Wu. "Degradation model and attention guided distillation approach for low-resolution face recognition". *Expert Systems with Applications*, 237:121536, 2024. doi: 10.1016/j.eswa.2023.121536.

[43] Mohsin Ullah. "Sub-center learning with distillation loss". Technical report, Capital University of Science and Technology (CUST), 2025. Available at https://cust.edu.pk/wp-content/uploads/2025/03/Mohsin_Ullah_EE.pdf.