AN AUTOMATED FACE DETECTION AND RECOGNITION SYSTEM FOR VIDEO SURVEILLANCE

A thesis submitted to the

UPES

For the Award of

Doctor of Philosophy

In

Computer Science & Engineering

By

SHAHINA ANWARUL

September 2023

Supervisor

Dr. Tanupriya Choudhury

External Supervisor

Dr. Susheela Dahiya

School of Computer Science

UPES

Energy Acres, P.O. Bidholi via Prem Nagar,

Dehradun, 248007: Uttarakhand, India.

AN AUTOMATED FACE DETECTION AND RECOGNITION SYSTEM FOR VIDEO SURVEILLANCE

A thesis submitted to the

UPES

For the Award of

Doctor of Philosophy

In

Computer Science & Engineering

By

SHAHINA ANWARUL

(SAP ID: 500057717)

September 2023

Internal Supervisor

Dr. Tanupriya Choudhury

Ex-Professor, UPES & Professor, Symbiosis International University

External Supervisor

Dr. Susheela Dahiya

Associate Professor, Graphic Era Hill University

School of Computer Science

UPES

Energy Acres, P.O. Bidholi, via Prem Nagar,

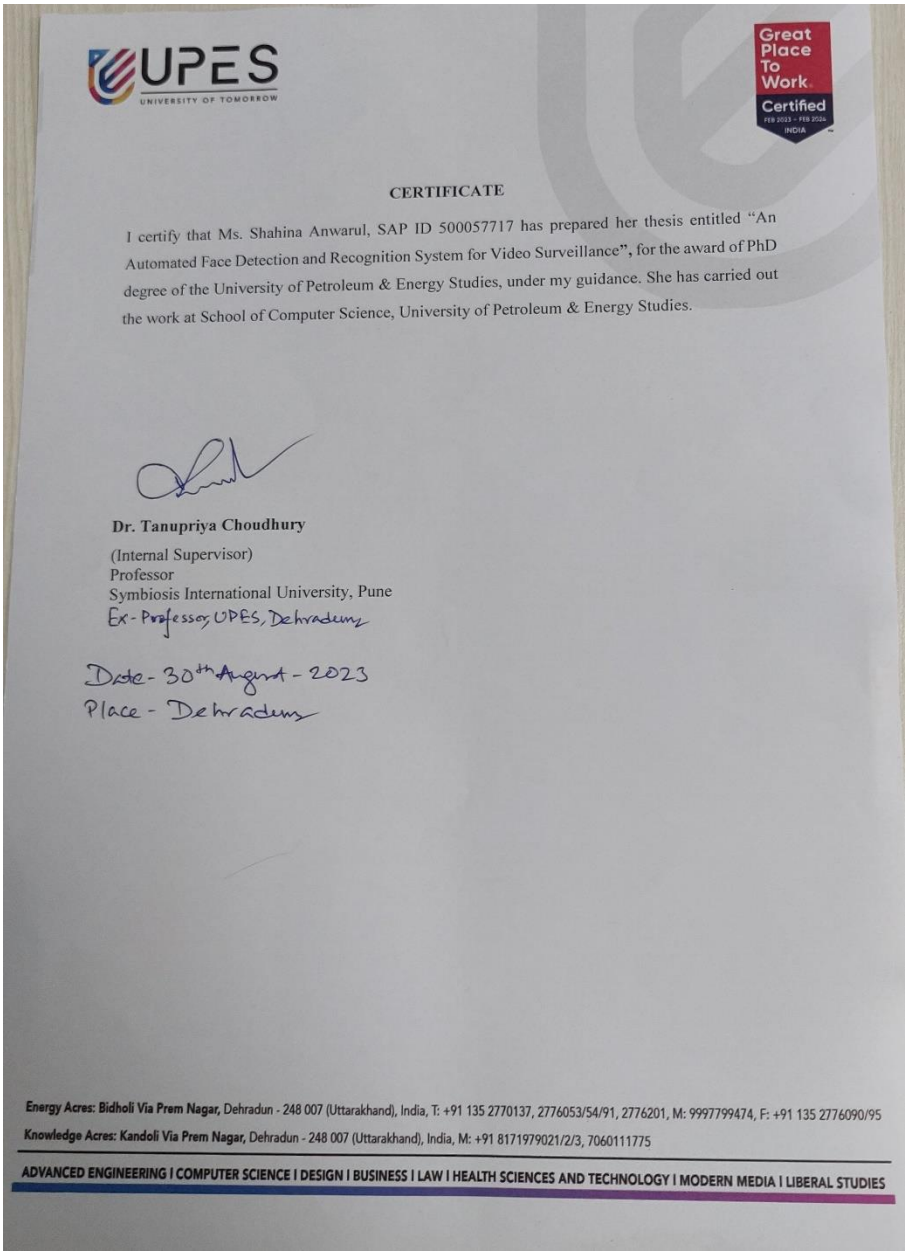
Dehradun, 248007: Uttarakhand, India

DECLARATION

I declare that the thesis entitled "**An Automated Face Detection and Recognition System for Video Surveillance**" has been prepared by me under the guidance of Dr. Tanupriya Choudhury, Ex-Professor, UPES & Professor, Symbiosis International University and Dr. Susheela Dahiya, Associate Professor, Department of Computer Science and Engineering, Graphic Era Hill University. No part of this thesis has formed the basis for the award of any degree or fellowship previously.

SHAHINA ANWARUL
School of Computer Science
UPES
Energy Acres, P.O. Bidholi, via Prem Nagar,
Dehradun, 248007: Uttarakhand, India

CERTIFICATE



UPES
UNIVERSITY OF TOMORROW

Great Place To Work. Certified FEB 2023 - FEB 2024 INDIA

CERTIFICATE

I certify that Ms. Shahina Anwarul, SAP ID 500057717 has prepared her thesis entitled "An Automated Face Detection and Recognition System for Video Surveillance", for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. She has carried out the work at School of Computer Science, University of Petroleum & Energy Studies.

Dr. Tanupriya Choudhury
(Internal Supervisor)
Professor
Symbiosis International University, Pune
Ex - Professor, UPES, Dehradun

Date - 30th August - 2023
Place - Dehradun

Energy Acres: Bidholi Via Prem Nagar, Dehradun - 248 007 (Uttarakhand), India, T: +91 135 2770137, 2776053/54/91, 2776201, M: 9997799474, F: +91 135 2776090/95
Knowledge Acres: Kandoli Via Prem Nagar, Dehradun - 248 007 (Uttarakhand), India, M: +91 8171979021/2/3, 7060111775

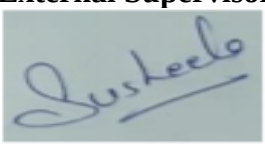
ADVANCED ENGINEERING | COMPUTER SCIENCE | DESIGN | BUSINESS | LAW | HEALTH SCIENCES AND TECHNOLOGY | MODERN MEDIA | LIBERAL STUDIES

CERTIFICATE

Graphic Era
HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

I certify that the thesis entitled ***"An Automated Face Detection and Recognition System for Video Surveillance"*** by **Shahina Anwarul (SAP ID: 500057717),** research scholar at UPES, Dehradun, submitted thesis in partial completion of the requirements for the award of the Degree of Doctor of Philosophy in School of Computer Science is an original work carried out by her under my supervision and guidance. It is certified that the work has not been submitted anywhere else for the award of any other diploma or degree of this or any other University.

External Supervisor



Dr. Susheela Dahiya
Associate Professor
Department of Computer Science and Engineering
Graphic Era Hill University, Dehradun.

ABSTRACT

An all-encompassing automated system for video surveillance, dedicated to recognizing faces, consists of various elements: face detection, face alignment, face recognition, and alert generation. In today's world, face recognition has become a powerful technology utilized in numerous applications, particularly in criminal identification. The ongoing manual examination of surveillance videos is an arduous process that demands significant visual focus but lacks mental engagement, making it prone to mistakes. Therefore, this research presents an automated facial recognition system as a solution to tackle these obstacles.

The current study consisted of three distinct phases. Initially, we conducted an evaluation of multiple existing face detection algorithms. After careful analysis, we determined that the Single-Shot Multibox Detector (SSD) is the most optimal method due to its superior speed and accuracy compared to other alternatives. In the following phase, we introduced a new model for face recognition based on ensemble learning. Recognizing faces has proven challenging due to factors such as pose variations, changes in lighting, aging effects, partial occlusion, and low resolution. Contemporary approaches to face recognition have limitations when dealing with these unconstrained conditions. Therefore, improving face recognition requires incorporating diverse deep learning architectures. Despite advancements in traditional deep learning techniques for face recognition systems, there is still a need for a robust and efficient solution. To address this gap, the research work has been proposed and implemented a Hybrid Ensemble Convolutional Neural Network (HE-CNN) model. This model is established through ensemble transfer learning from modified pre-trained models and contributes to achieving higher accuracy in face recognition tasks.

The model undergoes a two-phase training approach, incorporating a differential learning rate based on a one-cycle policy. This method greatly improves

the model's ability to recognize faces. It should be noted that these enhancements result in State-of-the-Art performance. To achieve this, the concatenation of Global Max Pooling (GMP) and Global Average Pooling (GAP), Batch Normalization (BN), a Fully Connected (FC) layer, and dropout are integrated into the classification layers of pre-trained models. The incorporation of these suggested modifications and refining of the training process, we observed outstanding results with a significant increase in recognition accuracy. An ablation study further confirms the positive impact of these changes on recognition accuracy. Additionally, extensive experiments have been conducted to evaluate the performance of the proposed HE-CNN model using benchmark datasets. The proposed and implemented model has been evaluated using a self-curated criminal dataset to demonstrate its real-time applicability in practical scenarios. Through careful parameter selection and customization of layers, the designed model achieved remarkable accuracy rates: 99.35% on Labeled Faces in the Wild (LFW), 91.58% on Cross Pose LFW (CPLFW), 99.63% on Georgia Tech (GT face), 99.21% on YouTube Faces (YTF), and 95% on the self-curated dataset. Lastly, in the presented work, an automated alert system has been created that identifies crime-prone areas and helps prevent criminal activities. This is done through the analysis of data obtained from the identification of criminals. The system proactively alerts law enforcement personnel about high-risk areas so they can be prepared and vigilant before any crimes occur. Alerts are sent promptly when individuals with criminal records are detected in specified regions.

In a gist, the present research, “An Automated Face Detection and Recognition System for Video Surveillance”, provided an efficient face recognition system based on the hybrid model. The hybrid model leverages the benefits of deep ensemble transfer learning techniques to construct a fast and highly accurate model. The novelty of this research work lies in balancing the trade-off between recognition accuracy and computational efficiency. In existing research, there is a

trade-off between accuracy and computation. Techniques that achieve high accuracy are expensive in terms of computational resources, while others prioritize computational efficiency at the cost of accuracy. The present research introduces a solution that overcomes this trade-off by presenting a computationally efficient model that maintains a high level of accuracy.

ACKNOWLEDGMENT

Ph.D. journey is truly a life-changing experience and a confluence of multiple learning for me. It is not possible to do without the support and guidance that I received from many people. I am extremely thankful to all of them. I express my heartfelt gratitude to my research supervisors Dr. Tanupriya Choudhury and Dr. Susheela Dahiya for their unconditional support and motivation during the entire research program. Their guidance always lent a hand to me not only in research but in real life also. Their simplicity, positivity, honesty, and discipline inspire me in every second of my life. I wish and try to be as honest and disciplined as they are, for my entire life. Without their continuous support, guidance, and constant feedback, this Ph.D., would not have been achievable.

I express my sincere gratitude to the Honorable Chancellor Dr. Sunil Rai, Honorable Vice Chancellor Dr. Ram Sharma, Registrar Mr. Manish Madan, Dean of the School of Computer Science Dr. Ravi S. Iyer. I am deeply thankful for the resources, opportunities, and academic freedom provided by UPES under your leadership. I greatly acknowledge all the panel members and reviewers who helped me in my research work with their valuable suggestions.

My deep appreciation goes to Mr. Abhishek Yadav, Dr. Mohammad Yaqoot, Dr. Deepa Joshi, Dr. Divya Srivastava, Dr. Sunil Kumar, Mr. Vidyanand Mishra, Ms. Tripti Misra, Dr. Neelu Jyoti Ahuja, and Ms. Diksha Chauhan for their much needed support during this research program. They were always helpful and provided me with their assistance throughout my research journey.

I am blessed with a beautiful and supportive family, I am unable to express my gratitude in words to my father Late Anwarul Haq Siddiqui, my mother Mrs. Gufrana Siddiqui, elder sister Ms. Shimaila Anwarul, and younger brother Mr. Noorul Haq Siddiqui for always believing in me and encouraging me to follow my dreams.

Most importantly, I express my gratitude to the Almighty for blessing me with the wisdom, good health, and resilience needed to undertake this research endeavor and to shower the blessings on me to complete my thesis.

SHAHINA ANWARUL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

Acronym	Meaning of Abbreviation
AI	Artificial Intelligence
ANNs	Artificial Neural Networks
API	Application Programming Interface
ASPP	Atrous Spatial Pyramid Pooling
BN	Batch Normalization
CCM-CCN	Cross-Correlation Matching CNN
CCTV	Closed-Circuit Television
CFR-CNN	Canonical Face Representation CNN
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CONV	Convolutional
CPLFW	Cross-Pose LFW
CPUs	Central Processing Units
CV	Computer Vision
DA	Domain Adaptation
DCNN	Deep CNN
DeCAF	Deep Convolutional Activation Feature
DPI	Dots Per Inch
DSN	Deeply-Supervised Nets
EBGM	Elastic Bunch Graph Matching
EV-SIFT	Entropy-based Volume SIFT
FC	Fully Connected
FCNT	Fully Convolutional Network-based Tracker
FERET	Facial Recognition Technology
FNR	False Negative Rate
FPR	False Positive Rate
FR	Face Recognition
FRNN	Full Resolution Residual Networks
GAN	Generative Adversarial Networks
GAP	Global Average Pooling
GMP	Global Max Pooling
GPS	Global Positioning System
GPUs	Graphical Processing Units

Acronym	Meaning of Abbreviation
GT	Georgia Tech
HE-CNN	Hybrid Ensemble Convolutional Neural Network
ILSVRC	ImageNet Large Scale Visual Recognition Competition
JPEG	Joint Photographic Experts Group
KDD	Knowledge Discovery in Databases
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LFW	Labelled Faces in the Wild
LRF	Learning Rate Finder
MDC	Minimum Distance Classifier
MDR-TL	Mean Distance Regularized Triplet Loss
ML	Machine Learning
MLP	Multilayer Perceptron
MTCNN	Multitask Cascaded Convolutional Neural Networks
NIN	Network in Network
NR	Not Reported
PaSC	Point and Shoot Face Recognition Challenge
PC	Proposed Classifier
PCA	Principal Component Analysis
PSCL	Point-to-Set Correlation Learning
PUB-FIG	Public Figures
RCNN	Recurrent CNN
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic
ROIs	Regions of Interest
RPNs	Region Proposal Networks
RU	ResNet Unit
S2S	Still-to-Still
S2V	Still-to-Video
SD	Standard Deviation
SGD	Stochastic Gradient Descent
SIFT	Scale Invariant Feature Transform
SIP	Single Image of Person

Acronym	Meaning of Abbreviation
SL	Super Learner
SOTA	State-of-the-Art
SSD	Single-Shot Multibox Detector
SVD	Singular Value Decomposition
TBE-CNN	Trunk-Branch Ensemble CNN
TPR	True Positive Rate
3D	Three Dimensional
V2V	Video-to-Video
VS	Video Surveillance
YTF	YouTube Faces

CHAPTER-1

INTRODUCTION

Biometric systems aim to authenticate individuals by utilizing one or multiple distinctive biometric characteristics, such as facial features, iris patterns, fingerprints, and other similar traits. The biometric traits can be classified into behavioral and physiological traits, as shown in Figure 1.1. Traditional authentication methods, such as identification cards and passwords, are often lost or stolen, while biometric-based systems improve security over traditional methods. Broadly, biometric applications can be categorized into three primary categories: verification, identification, and screening. Verification involves comparing an individual's biometric data with the stored data to validate their identity (referred to as one-to-one matching). The second category entails comparing an individual's biometric traits with the traits of various individuals stored in the system (known as one-to-many matching). In the last category, a small number of target persons are matched with unknown persons from a large group of people (*i.e.*, many-to-some matching).

There is a growing need for biometric security solutions to protect against fraud, theft, and other risks. Face Recognition (FR) holds a crucial position in biometrics-based security techniques and has proven to be a valuable tool across a diverse range of applications, including disease diagnosis, forensic analysis, secure transactions, age estimation, missing person searches, e-passport identification, mask recognition, and more [1] [2] [3]. Face Recognition has drawn the most attention from researchers among the many biometric applications in recent years since it is more covert, non-intrusive, and requires less human involvement than other biometrics like the iris, fingerprint, or palmprint. The FR process involves analyzing and comparing essential facial features and expressions, aiming to enhance the intelligence and safety of our world. This technology finds applications

in authentication and surveillance, allowing for the identification of individuals and, when needed, the detection of suspicious behavior or suspects. In surveillance applications, Face Recognition is a crucial component for person identification. The automated FR system encompasses fundamental steps such as face detection, face alignment, face recognition, and alert generation, as depicted in Figure 1.2.

- **Face Detection:** Identify the faces in an image or video using landmarks on the face such as eyes, nose, mouth, etc.
- **Face Alignment:** Alignment and normalization of faces for better recognition accuracy.
- **Face Recognition:** Recognize a specific person by comparing an image or video with the stored dataset.
- **Alert Generation:** Send an alarm message to the concerned person to reduce human intervention to detect people.

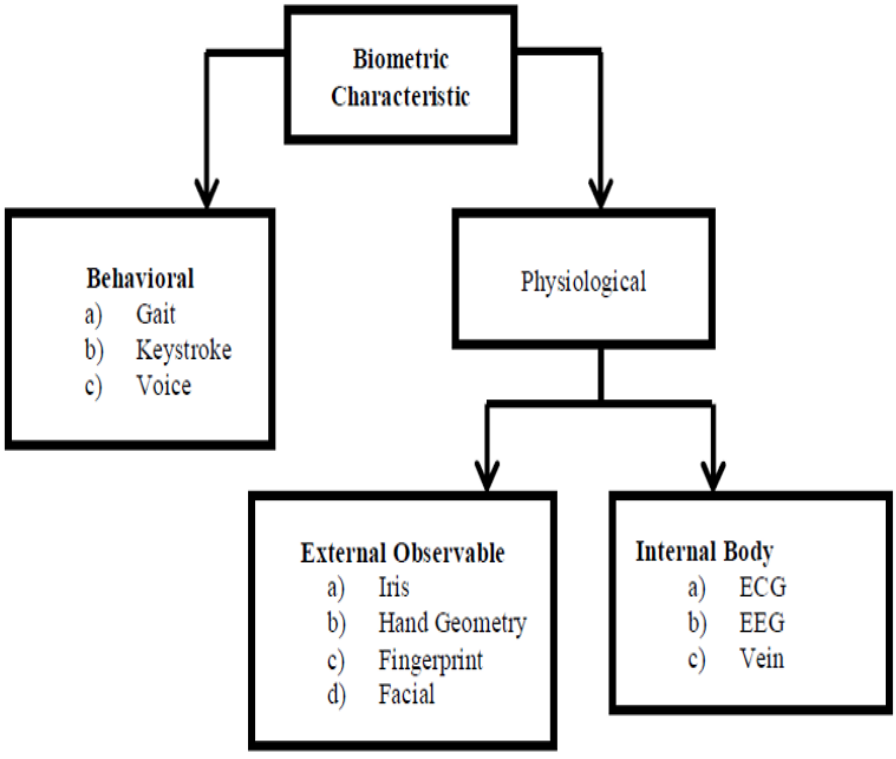


Figure 1.1 Classification of Biometric Characteristics

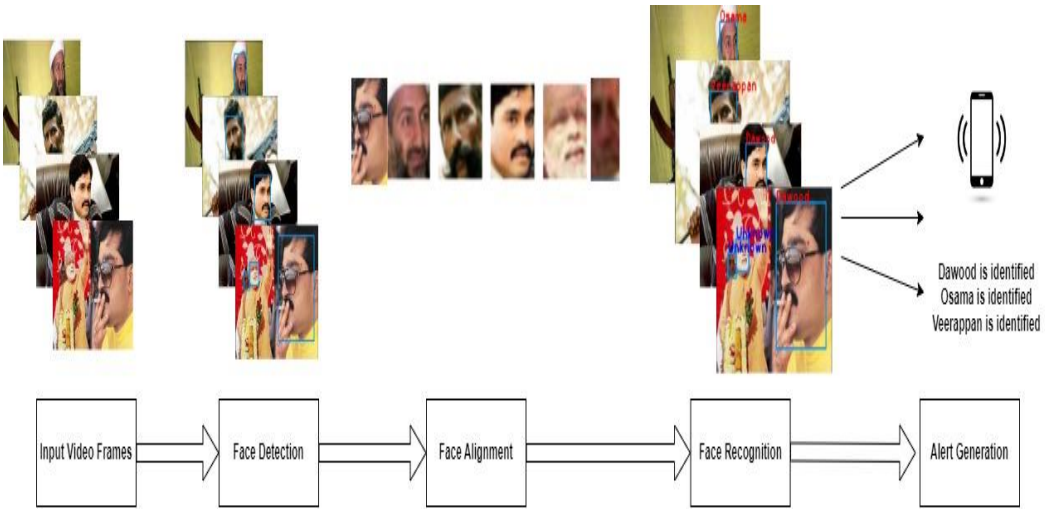


Figure 1.2 The Block Diagram of an Automated FR System

The task of FR in photos and videos is certainly difficult, and reaching 100% accuracy is a constant endeavor due to different factors influencing FR system performance. Despite intensive efforts, sufficient results have yet to be obtained, owing mostly to the numerous factors influencing the accuracy of these systems. Numerous studies have found that occlusion, low resolution, noise, illumination, position change, face expression, aging, and plastic surgery have an impact on recognition accuracy [4] [5] [6]. These components are divided into two categories: internal and extrinsic factors [4]. Intrinsic factors are the physical qualities of the human face that affect recognition accuracy, such as aging, facial expression, and plastic surgery. Extrinsic factors, on the other hand, alter the facial appearance and include occlusion, low resolution, noise, lighting, and position change, as shown in Figure 1.3. Depending on the nature of the training and test data, Zhao *et al.* [7] and Tan *et al.* [8] offered three basic scenarios for creating and evaluating FR systems. These are Still-to-Still (S2S), Still-to-Video (S2V), and Video-to-Video (V2V) FR scenarios, as depicted in Figure 1.4. In the S2S scenario, the FR system utilizes Regions of Interest (ROIs) extracted from still images of specific subjects as reference data to build a face model during the registration phase. Subsequently, the system performs real-time recognition using other still images as operational

data. In the S2V scenario, ROIs from reference still images are used to build face models, but the system operates on video streams for detection purposes. Lastly, the V2V scenario utilizes frames extracted from video streams as dual-purpose data, serving both as reference and operational inputs for Face Recognition [9]. The S2V FR encounters challenges due to environmental differences between the source (registration) and destination (surveillance) domains. The captured images used during registration were obtained under controlled conditions. In contrast, the images captured by surveillance cameras are subject to unconstrained factors, such as low resolution, occlusion, lighting variations, and more. Face Recognition systems tailored for Video Surveillance (VS) purposes strive to precisely detect and recognize individuals of interest across a distributed network of cameras.

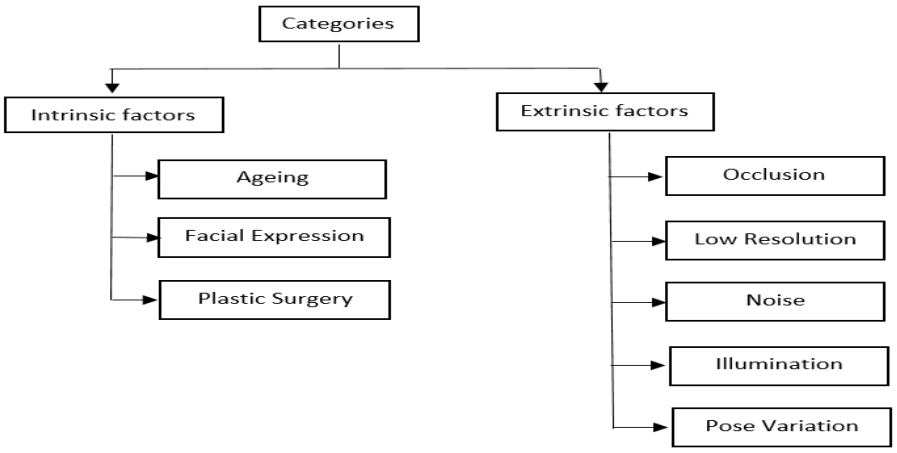


Figure 1.3 The Classification of the Factors Affecting FR Accuracy

(a) S2S (b) S2V (c) V2V

Figure 1.4 Different Scenarios for the FR System

Extensive research has been dedicated to various face detection and recognition techniques. Traditional approaches primarily involve the use of Principal Component Analysis (PCA) for Face Recognition, achieving accuracy rates ranging from 69% to 95% in controlled environments [10]. PCA has also been combined with other methods such as Singular Value Decomposition (SVD) and Fisherface techniques, resulting in recognition rates of 93.92% and 99.5% for frontal faces [11]. Furthermore, researchers also investigated non-frontal FR techniques, such as mirroring, fitting, stretching, segmentation, and Three-Dimensional (3D) operations [12] [13]. However, the effectiveness of these methods tends to decline when facial images are captured in challenging environmental conditions, such as inadequate lighting, low-resolution cameras, and occluded facial images [14] [15] [16]. With the recent advent of deep learning [17] [18], the limitations of traditional methods have come to an end [19]. However, the dependency on the enormous amount of data and systems with high computing power (*e.g.*, parallel processing systems accelerated with Graphical Processing Units (GPUs)) are still the challenges of deep learning techniques [20] [21]. The large amount of annotated facial datasets for the FR tasks is difficult to obtain due to the privacy concerns of the individuals [22]. The recommended solution to address these challenges is deep ensemble transfer learning [23]. It saves our time and resources. Transfer learning is a technique for using the feature representation from a pre-trained model. Building and training a model from scratch is a tedious procedure. Instead of this lengthy process, transfer learning uses the weights from the pre-trained architectures to train the new model for the desired task [24]. Ensemble learning is employed to enhance recognition accuracy by averaging the weights of multiple deep-learning models. It combines the benefits of deep learning and ensemble learning to achieve improved generalization performance in the final model [25]. The main objective of this research is to develop an efficient system that enables the recognition of faces using minimal facial data and computational resources while maintaining high accuracy. Therefore, we leveraged the concept of

ensemble transfer learning to introduce a highly efficient FR system based on deep learning. In the available research articles [26] [27] [28], a trade-off is observed between accuracy and computation. Certain articles achieved high accuracy but at the expense of computational resources, while others prioritized computational efficiency at the cost of accuracy. The proposed and implemented research introduces a solution that overcomes this trade-off by presenting a computationally efficient model that maintains a high level of accuracy.

1.1 Motivation

Face detection and recognition play a crucial role in authentication systems based on biometric data, serving purposes in both authentication processes and surveillance. As scams and fraudulent activities continue to rise, facial recognition has become an essential system for ensuring security. Extensive research has been conducted globally to advance this field; however, despite continuous efforts, there is still a lack of robust and effective automated systems capable of performing well in both controlled and uncontrolled environments. Face Recognition has always been a highly intricate and demanding task, as it strives to replicate the human ability to perceive and identify faces. Nonetheless, human capabilities have limitations when dealing with various ambiguous phenomena. Hence, there is a need for an automated electronic system with high recognition accuracy and fast processing capabilities. The demand for biometric security systems has witnessed a substantial surge in recent times, driven by the need for enhanced protection and security against fraud, theft, and other related threats. Among the various biometric-based systems, Face Recognition has emerged as a prominent and effective solution. It serves various applications, including forensics, criminal identification, surveillance, and fraud prevention, as it can authenticate an individual's identity and recognize individuals in different scenarios. Face Recognition system is used in banks, railway stations, airports, and other public places as a security control system where Closed-Circuit Television (CCTV) cameras are leased to identify

individuals. It is also used in other sectors such as education, healthcare, media and entertainment, *etc.*, as illustrated in Figure 1.5 [29]. Video Surveillance recordings can be used to identify the suspect at the crime scene. Monitoring surveillance videos continuously is a very tiring task that requires visual attention and is also boring, leading to more opportunities for error. Automated surveillance that uses an intelligent system to monitor activities and raise an alarm, when necessary, can form an effective security system. In these real-time scenarios, there is a high possibility that the captured image has a large pose variation, faces are obscured by glasses, clothes, *etc*., the lighting effect of the image might be dark, the facial expression might be different, *etc*. These are the factors that contribute to the deterioration in facial recognition accuracy. Therefore, an effective automated facial recognition system that offers high accuracy with minimal computational cost is the need of the hour.

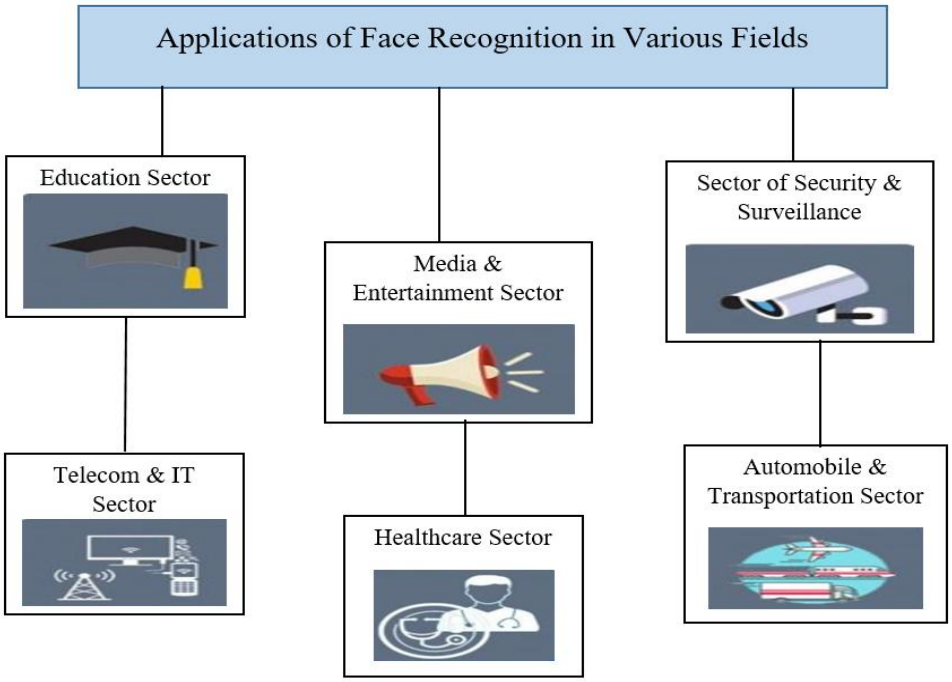


Figure 1.5 Applications of FR in Various Sectors

1.2 Problem Description

The reactive Video Surveillance systems wait for the event to occur, and later analysis is done. Therefore, there is a need for a proactive automated system that offers higher recognition accuracy and faster processing to prevent losses resulting from abnormal events. In real-time Video Surveillance systems, face detection and recognition play a challenging role due to the presence of various unconstrained factors such as pose variation, partial occlusion, illumination, and low resolution.

Existing deep learning-based FR systems necessitate a substantial amount of meticulously cleaned and labeled face images for effective training and feature learning. Consequently, the training phase of these FR models demands significant computational resources and is time-consuming [26] [30] [31]. Moreover, during the operational phase, numerous pre-processing steps are performed on captured face images, requiring substantial computing power and time. As a result, these approaches are not suitable for real-time applications like Video Surveillance [32]. Additionally, existing FR systems are reactive in nature, conducting analysis only after an individual has been encountered [33] [34] [35]. Prior research has focused either on developing FR models [26] [30] [31] or directly implementing existing models into systems [33] [34] [36]. As far as our knowledge extends, there has been no comprehensive automated research-oriented FR system proposed to address these challenges.

1.3 Objective

Analysis and design of a proactive and efficacious Face Recognition system to mitigate the effects of factors like pose, partial occlusion, illumination, and low resolution that degrade the facial recognition accuracy for Video Surveillance.

Sub-objectives

1. To identify an efficient face detection algorithm.
2. To propose a Hybrid Ensemble Convolutional Neural Network (HE-CNN) model for Face Recognition and analyze the performance thereof.
3. To develop an automated FR system that does not require human intervention for the identification of an individual.

1.4 Thesis Contribution

The several noteworthy contributions of the proposed research are summarized below.

1. The proposed modified architecture of the baseline model

(a) Instead of training the Convolutional Neural Network (CNN) model from scratch, we adopted the concept of transfer learning to obtain fine-tuned baseline models for addressing the discussed problem.

(b) The architecture of the baseline models (VGG19, DenseNet169, and ResNet50) is enhanced by incorporating Global Average Pooling (GAP) and Global Max Pooling (GMP), a Fully Connected (FC) layer, Batch Normalization (BN), and dropout in the classification layer. These modifications resulted in the State-of-the-Art (SOTA) competent results.

(c) Two-phase training has been implemented, involving the freezing and unfreezing of model layers, along with the optimization of hyperparameters. This approach significantly enhanced the accuracy of Face Recognition.

2. A novel optimized HE-CNN model is proposed and implemented that uses the average weighting of modified fine-tuned baseline models to improve the recognition rate.

3. In the designed system, the assessment of the Single-Shot Multibox Detector's (SSD) [37] performance is done for the face detection module and contrasted with the Multitask Cascaded Convolutional Neural Networks (MTCNN) [38]

framework, the Haar feature-based cascade classifier [39], and the Local Binary Pattern (LBP) feature-based cascade classifier [40] framework. Two widely used datasets, Labeled Faces in the Wild (LFW) [41] and Cross-Pose LFW (CPLFW) [42], as well as a self-curated collection of mugshots, are used in this study.

4. The proposed changes in the architecture and the techniques used for hyperparameter optimization have been demonstrated using image benchmark datasets such as Georgia Tech (GT) face [43], LFW, CPLFW, and self-curated dataset, as well as benchmark video dataset such as YouTube Faces (YTF) [44].

5. An unprecedented approach is designed to make the system intelligent that reduces human intervention. It has two modules: one that predicts crime-prone locations and the other that generates alerts. The automated facial recognition system has been implemented in criminal recognition to demonstrate the real-time application of the presented research.

6. The images in standard datasets typically contain only a single face per image, while real-time scenarios often involve multiple faces in a single image. To address this, we also developed a self-curated dataset consisting of mugshots. This dataset includes images of 10 criminals collected from freely available sources on the Internet. The purpose of this dataset is to showcase the practical application of the proposed recognition system in real-time scenarios. The dataset was created considering the factors present in the real-time scenario, such as low resolution, partial occlusion, lighting, *etc.* The self-curated dataset is available for research at the link provided: https://data.mendeley.com/datasets/226275vfxz/2.

1.5 Thesis Outline

The structure of the remaining thesis is organized into six chapters, including the introduction chapter. In Chapter 2, a comprehensive exploration of State-of-the-Art Face Recognition research is undertaken, starting with traditional algorithms and progressing to advanced deep learning-based approaches, transfer learning-

based methods, and ensemble learning-based techniques. The chapter also addresses the challenges existing in Face Recognition and discusses standard datasets used to evaluate FR algorithms.

In Chapter 3, both the self-curated mugshot dataset and the selected standard datasets are discussed in detail. To emphasize the impact of unconstrained factors, datasets captured in both constrained and unconstrained environments are utilized. An algorithm for data oversampling is introduced in this chapter as part of an effort to ensure that a balanced dataset is used in evaluating the proposed work.

Chapter 4 introduces a novel method for an automated FR system. Deep ensemble transfer learning is used in the proposed system to strike a balance between accuracy and computational resources. In the proposed and implemented system, face detection is handled by SSD, while Face Recognition is handled by a hybrid model. The suggested FR system also includes alert generation to reduce human intervention in recognizing individuals. This chapter further elaborates on the concept of two-phase learning, which is used to train the proposed modified models.

In Chapter 5, the essential hyperparameters for fine-tuning the model are discussed in detail. Furthermore, the suggested architecture's performance is tested and compared to existing approaches to illustrate its superiority. The proposed work is evaluated using various metrics such as accuracy, precision, recall, error rate, and Receiver Operating Characteristic (ROC) curve. In this chapter, an ablation study is conducted to evaluate the impact of the suggested modifications to the pre-trained models. At the end of the chapter, the execution time analysis of the proposed system is discussed.

Finally, in Chapter 6, the thesis is summarized, conclusions are drawn, and future research directions are explored. The diagrammatic representation of the thesis outline is given in Figure 1.6.

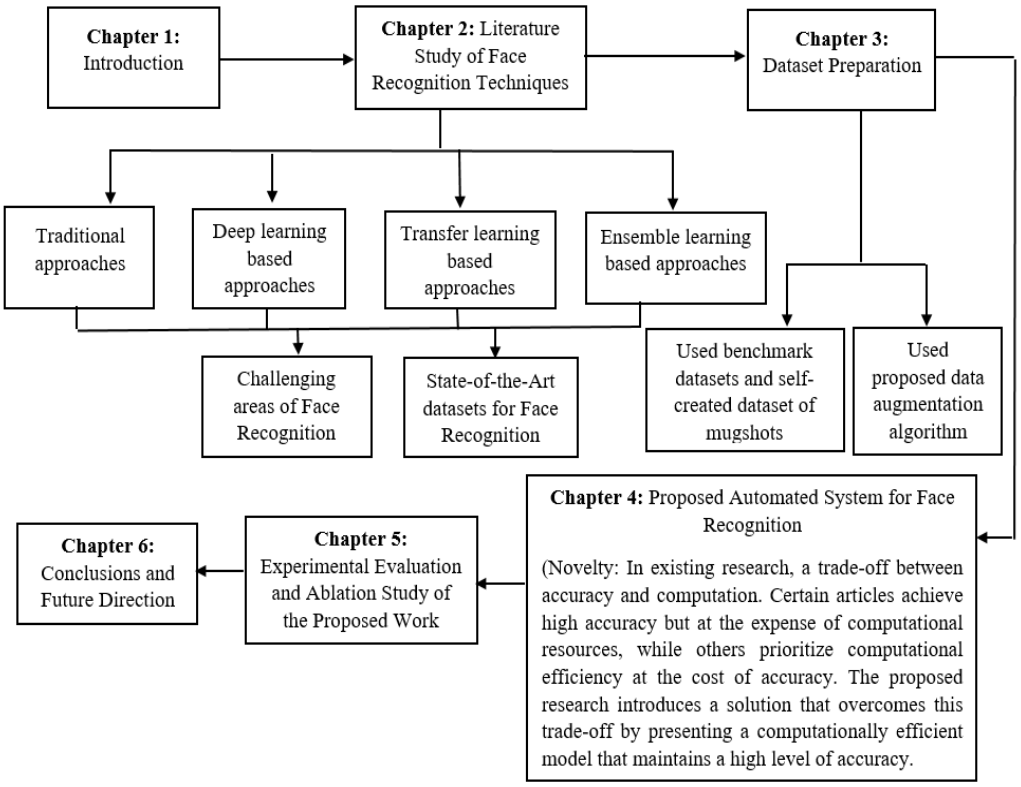


Figure 1.6 The Diagrammatic Representation of the Organization of the Thesis

1.6 Summary of the Chapter

This chapter begins with an overview of the intended topic, followed by a motivation for the proposed research, then goes on to describe the challenges of the study and summarize the contributions carried out by this thesis. Furthermore, a thesis outline describes the flow of the proposed research. The next chapter elaborates on the existing State-of-the-Art approaches proposed by various researchers for face recognition tasks.

CHAPTER-2

LITERATURE STUDY

Computer Vision (CV), a specialized domain within Artificial Intelligence (AI), empowers computers and systems to extract insights from digital images, videos, and other visual inputs. Subsequently, these insights are utilized to execute actions or formulate predictions. This interdisciplinary field bridges diverse areas of study, including computer science (theory, architecture, systems, algorithms), engineering (image processing, natural language processing, speech processing, robotics), biology (neuroscience), mathematics (machine learning, information retrieval), and physics (optics). The key elements of Computer Vision include visual recognition tasks such as image classification, object detection, localization, and segmentation. As the adoption of AI technologies reshapes numerous industries since the inception of machine learning, computer vision emerges as a pivotal player. Particularly in the realm of face recognition, Computer Vision provides algorithms and methodologies to scrutinize and process visual data embedded in images or videos containing human faces. This chapter delves into an intricate exploration of face recognition techniques, encompassing conventional, deep learning, transfer learning, and ensemble learning approaches. Moreover, it presents a comprehensive overview of the contemporary State-of-the-Art in the domain of face recognition.

2.1 Traditional Algorithms for Face Recognition

Numerous investigations have been undertaken to explore diverse methodologies for face detection, identification, and matching. Traditional algorithms for face recognition include Scale Invariant Feature Transform (SIFT) [45], Principal Component Analysis (PCA) [10] [46] [47], AdaBoost, Linear Discriminant Analysis (LDA), Elastic Bunch Graph Matching (EBGM) [48], Fisherface, and Singular Value Decomposition (SVD) [11]. However, these

approaches are susceptible to limitations stemming from variations in illumination, pose, and expression. Furthermore, their efficacy in recognizing faces is diminished in uncontrolled settings. Khan *et al.* [10] introduced an automatic face recognition system employing the PCA eigenface algorithm. Experimental outcomes showcased an 86% recognition accuracy within controlled environments and an 80% recognition accuracy within uncontrolled scenarios. Nonetheless, this system encounters challenges in delivering satisfactory outcomes for videos of low resolution and considerable pose deviations. In another study [48], a comprehensive analysis comparing the AdaBoost, PCA, LDA, and EBGM algorithms for face recognition was presented. This comparison highlighted their drawbacks, advantages, success rates, and other pertinent factors. Following the evaluation, PCA emerged with the highest success rate (85%–95%); however, it exhibited limitations when applied to video datasets. It is important to observe that the algorithms discussed are primarily well-suited for datasets that are smaller or less complex. Abdullah *et al.* [46] introduced a method for identifying criminals through facial recognition. This method employed PCA to identify criminal individuals based on their faces. The reported results indicated 80% accuracy in recognition; however, additional testing is necessary to validate the proposed approach. In a separate study, Dhamija *et al.* [11] utilized a blend of PCA, Fisherface, and SVD techniques for facial recognition. Their proposed system underwent testing on the AT&T face dataset, achieving an estimated recognition rate of about 99.5% using the leaving-one-out technique and 93.92% using the hold-out approach. Kavitha *et al.* [12] proposed another technique for transforming non-frontal faces into frontal perspectives. This was accomplished by performing a series of fitting, mirroring, and stretching operations to generate a frontalized facial appearance. Their approach was tested on Facial Recognition Technology (FERET), LFW, and Public Figures (PUB-FIG) face datasets to demonstrate the accuracy of the suggested approach. It can, however, only handle pose variations up to $\pm 22.5^0$. The proposed face recognition system by Gao *et al.* [45] is based on a large number of virtual

views and alignment errors. The Lucas Kanade and SIFT algorithms were used in this technique. The proposed method performed well on the FERET dataset, outperforming conventional face recognition methods by approximately 38%. It is, however, limited to position alterations of up to 60 degrees, and recognition accuracy decreases beyond 40^0. Additionally, the time complexity of the proposed algorithm is higher compared to other comparative algorithms. Ahonen *et al.* [40] introduced a method using LBP for face representation. The method underwent testing on the FERET dataset, yielding a 97% accuracy rate when dealing with images exhibiting various facial expressions. However, its performance was suboptimal when handling other variables. In a separate study, Kakkar *et al.* [49] crafted a system for recognizing criminals through facial identification. Their approach hinged on a Haar feature-based cascade classifier and Local Binary Pattern histogram. Meanwhile, Sable *et al.* [50] introduced an innovative technique termed Entropy-based Volume SIFT (EV-SIFT) aimed at recognizing surgically altered faces. The system was evaluated for various types of plastic surgeries, and different recognition rates were achieved for each surgery type. These traditional face recognition algorithms reviewed in Table 2.1 have limitations when environmental conditions are not controlled, such as poor lighting, non-occluded images, and low-resolution cameras. The emergence of deep learning has overcome some of the constraints associated with traditional approaches.

Table 2.1 A Tabular Representation of Traditional Face Recognition Methods Used in Different Studies

S. No.	Authors	Year	Dataset	Algorithm Used	Recognition Accuracy	Research Gaps
1.	Khan *et al.* [10]	2018	NCR-IIT facial database & Real-time video stream	PCA Algorithm	69% - 86%	1) PCA exhibited limited performance in uncontrolled environment 2) The accuracy of FR is pivotal, as the entire systems' success hinges on it. 3) Fails to yield improved

						outcomes for low-resolution video and pose variation.
2.	Banerjee *et al.* [51]	2018	Point and Shoot Face Recogniti on Challenge (PaSC) videos & CW images, CMU Multi-PIE dataset	Supervised learning, Viola Jones, Generic 3D model	88.453%-97.282%	1) All the discussed frontalization methods experienced high failure rates beyond the 40-degree yaw angle. 2) Focused only on a pose and somewhere on illumination and expression. 3) Supervised learning approach is used so it requires clean and labeled training data.
3.	Abdullah *et al.* [46]	2017	Real-time video stream	PCA Algorithm	80%	1) The efficiency of PCA diminishes when handling video datasets. 2) PCA exhibits better performance when applied to frontal faces. 3) Other relevant factors are not considered. 4) No dataset used to evaluate the performance of the algorithm.
4.	Gao *et al.* [45]	2015	FERET	Lucas Kanade, SIFT, Two-phase alignment error	99.521 % for ± 15 degree	1) The proposed approach capable of addressing pose variations within a range of ± 60-degree. 2) Recognition accuracy deteriorates significantly when surpassing 40^0. 3) Generating multiple virtual views for all database images is not a practical endeavor.
			LFW		26% for ± 60 degree	
5.	Huang *et al.* [52]	2015	COX Face dataset	Point-to-Set Correlation Learning (PSCL)	50.961% - 53.263%	1) Unconstrained factors such as aging, occlusion and plastic surgery are not considered in the proposed dataset.

6.	Fathima *et al.* [53]	2015	AT&T, MIT-India and Faces94 datasets	Gabor wavelet and Linear Discriminant Analysis	88% - 94.024%	1) Does not work well for faces with different pose distribution.
7.	Lei *et al.* [54]	2009	CMU-MIT face dataset	Modest Adaboost, Improved Independent Component Analysis, Hausdorff distance	95.206%	1) Experiments are not performed to evaluate the recognition rate. 2) A large number of images required to train the system for the recognition stage.

2.2 Deep Learning-based Approaches for Face Recognition

This section discusses deep learning-based algorithms developed for face recognition. Before that, the next subsections introduce deep learning and explore various architectures developed from early to advance deep CNNs.

2.2.1 Deep Learning

The concept of deep learning helps to provide higher recognition accuracy for the classification models in comparison to traditional approaches. Deep learning is a subfield of Artificial Intelligence and Machine Learning (ML) that includes statistical analysis techniques that train data recursively in order to provide predictions, as depicted in Figure 2.1 [55]. The distinguishing feature of deep learning models is their ability to learn and improve automatically through experience, enabling them to make predictions on unfamiliar data [56]. Within the framework of Machine Learning, the identification of significant features that capture anomalies or patterns in the data holds utmost importance. These features were traditionally primarily created through human expertise. Nevertheless, models may now learn these features on their own though the advancement of machine learning techniques. Artificial Neural Networks (ANNs) serve as a widely embraced computational model in machine learning, aiming to mimic the learning process of the human brain. Neural

networks, also known as perceptrons, have been in existence since the 1940s but have gained prominence in the field of artificial intelligence over the past few decades. The development of a technique called backpropagation is a key factor propelling their prominence in the field of Machine Learning. Backpropagation facilitates the ability of neural networks to modify the weights within the hidden layer of neurons in accordance with the intended output [57].

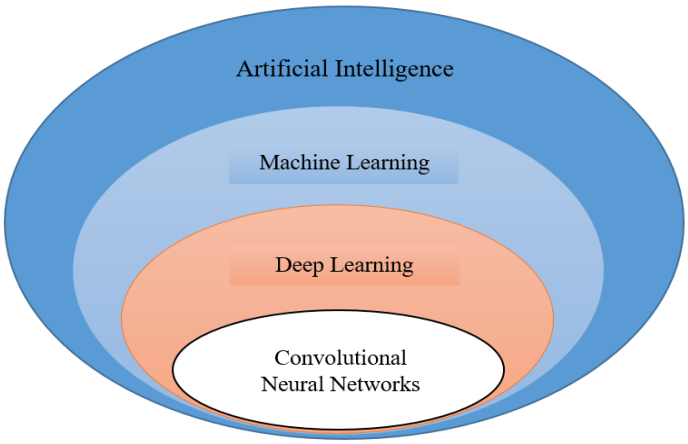


Figure 2. 1 Classification of Artificial Intelligence

Deep learning represents the advancement of Artificial Neural Networks, characterized by the incorporation of multiple hidden layers that enable higher levels of abstraction. The introduction of deep layers into the model has significantly enhanced the accuracy of task predictions by enabling the system to learn complex data [58]. A pivotal role in implementing deep learning-based approaches is played by CNNs [59] [60], which consist of convolutional layers, subsampling layers, and fully connected layers. The feature learning process involves the convolutional and subsampling layers, whereas the fully connected layer is used for classification, as illustrated in Figure 2.2. The emergence of CNNs has revolutionized feature learning techniques, as they have the ability to learn features automatically instead of relying on manual construction. CNNs have witnessed remarkable success in various computer vision tasks and are considered a significant breakthrough in machine

learning. One notable model, AlexNet, introduced by Krizhevsky *et al.* [61], brought about a paradigm shift in computer vision in 2012. AlexNet's architecture is similar to LeNet-5 [62], however it was first created to compete in the ImageNet competition. Its triumph in the ImageNet competition effectively demonstrated its efficacy, leading to widespread adoption within the computer vision community. Effective regularization parameters, data propagation techniques, rectified linear units, and the use of Graphics Processing Units (GPUs) to meet computing demands were all credited with this success. One of the top ten deep learning achievements in 2013 was AlexNet. The greatest strength of a CNN lies in its deep architecture which enables the extraction of sophisticated features at various levels of abstraction [61] [63].

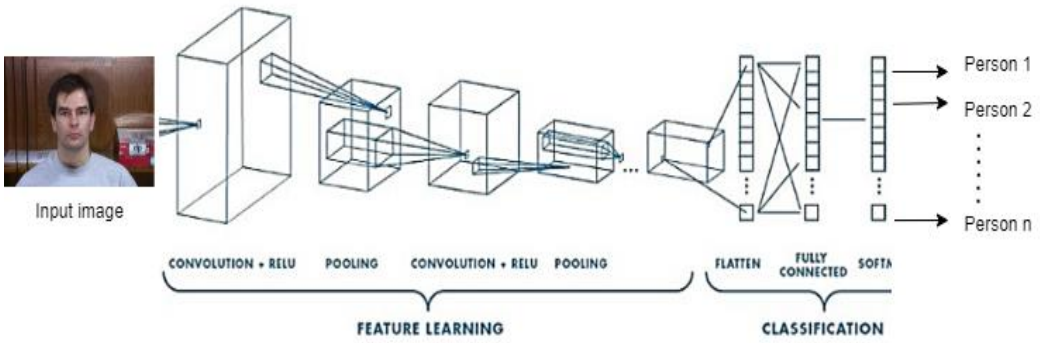


Figure 2.2 The Flow of the Working of CNN

2.2.1.1 Early Deep CNNs

The early deep CNNs first emerged in the late 1990s, starting around 1998. A CNN, also known as a ConvNet, stands as a distinctive and multi-tiered neural network deliberately designed for the task of pattern recognition. Its specialization lies in the capability to directly discern visual patterns from pixelated images, often requiring minimal to no preliminary data preprocessing. A sizable visual dataset created for use in image classification and object detection was made available by the ImageNet project [64]. In order to promote the development and assessment of cutting-edge algorithms, this project also ran the ImageNet Large Scale Visual

Recognition Competition (ILSVRC), an annual software competition [64]. The revolutionary CNN architecture LeNet-5 is presented in this section, followed by discussions of the leading CNN architectures of the ILSVRC: AlexNet, Network in Network (NIN), VGGNet, GoogLeNet, ResNet, and DenseNet. In this thesis, the collection of specified CNN architectures is referred to as L-A-N-V-G-R-D.

a) LeNet-5 (1998): Comparing conventional architecture to traditional neural networks has resulted in a series of advancements in image classification. LeNet-5 [62], the first CNN model released in 1998, had seven layers, only three of which were convolutional (C) and one of which was Fully Connected (FC), with a total of 60,000 parameters. In Figure 2.3, this network is displayed. The output of this network is a digit between 0 and 9, which is used to classify and identify 32 x 32-pixel grayscale handwritten numerals.

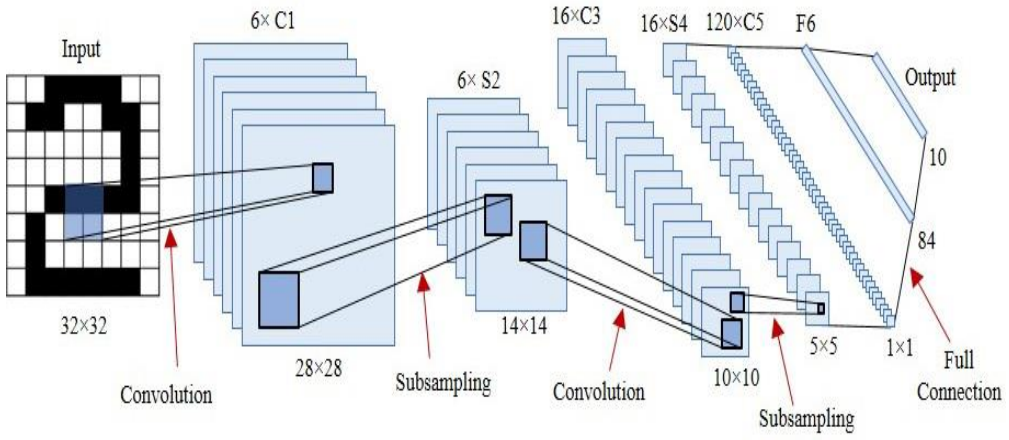


Figure 2.3 LeNet-5 Architecture consisting of 7 Layers [62]

b) AlexNet (2012): Higher-resolution images need to be processed using larger convolutional layers. Thus, AlexNet, which had 60 million characteristics in five convolution layers and three fully connected layers, is credited with starting the background of deep learning [61]. Figure 2.4 depicts the AlexNet architecture. The reasonably quick and simple AlexNet is slightly changed into ZF-Net [65]. This network performed substantially better than its predecessors [61] [62]. In a

conventional classification network, AlexNet has been applied after downsizing the input image and applying convolutional and FC layers. The output would then be the expected class label for the input image.

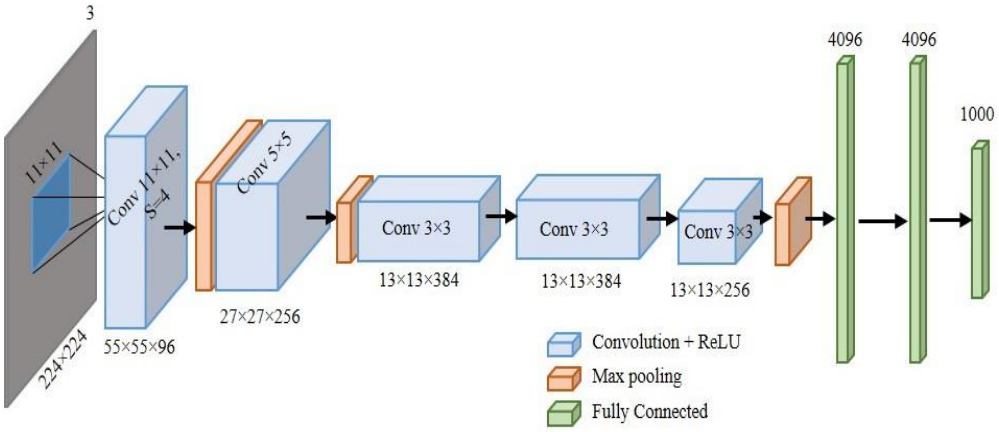


Figure 2.4 AlexNet Architecture [61]

c) NIN (2013): The capacity to distinguish between local patches within the input patch was improved by a Network in Network (NIN) design [66]. Three micro neural networks, essentially nonlinear function approximators, are stacked to generate this model. The Multilayer Perceptron (MLP) is used to create the tiny neural networks. As shown in Figure 2.5, the filter size for each layer of the MLP structure is 1x1, except for the first layer. Like CNN, the micro-networks are slid over the input to produce the feature maps, which are then supplied into the following layer. Multiple MLP structures are stacked to provide deep NIN, while the classification layer uses Global Average Pooling.

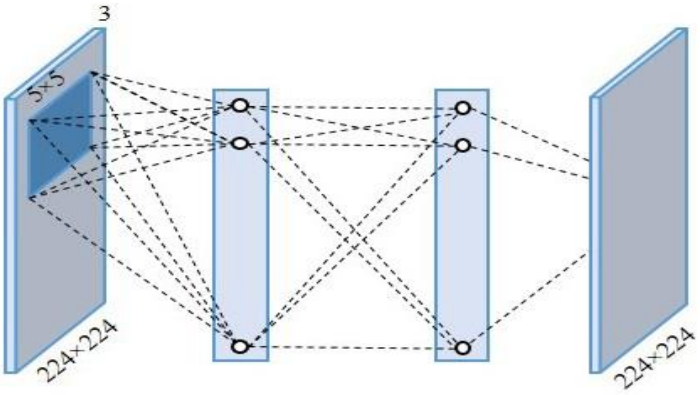


Figure 2.5 MLP Structure [66]

d) VGGNet (2014): This network's primary contribution is to assess correctness through deepening the network. This network, which had up to 19 layers and 138 million parameters, was made more accurate at classifying by using mini batch gradient descent with speed and dropout [67]. Six VGGNet configurations have been proposed, ranging from 11 weight layers (eight convolution and three fully linked layers) to 19 weight layers (with 16 convolution and three fully connected layers). The count of filters (depth) in each layer accumulates to 512, originating from an initial count of 64 in the first layer and progressively doubling after each max-pooling layer. Figure 2.6 depicts the VGGNet-16 design. Due to its extremely homogeneous design, VGGNet placed first in the single-object localization test at ILSVRC 2014 [64].

e) GoogLeNet (2015): The first section of the GoogLeNet design is similar to LeNet in Figure 2.3 and AlexNet in Figure 2.4, as shown in Figure 2.7, while the block's stack is derived from VGGNet in Figure 2.6. LeNet, AlexNet, and VGGNet's stack of FC layers are swapped out for GoogLeNet's worldwide mean pooling at the network's end. Google's top-5 error rate was 6.67%, which is quite near the level of human performance. It won first place in the ILSVRC 2014's classification and detection task [64]. The subsequent adoption of Batch

Normalization (BN) speeds up the training process for GoogLeNet [68]. Figure 2.8 shows the GoogLeNet model with 22 layers.

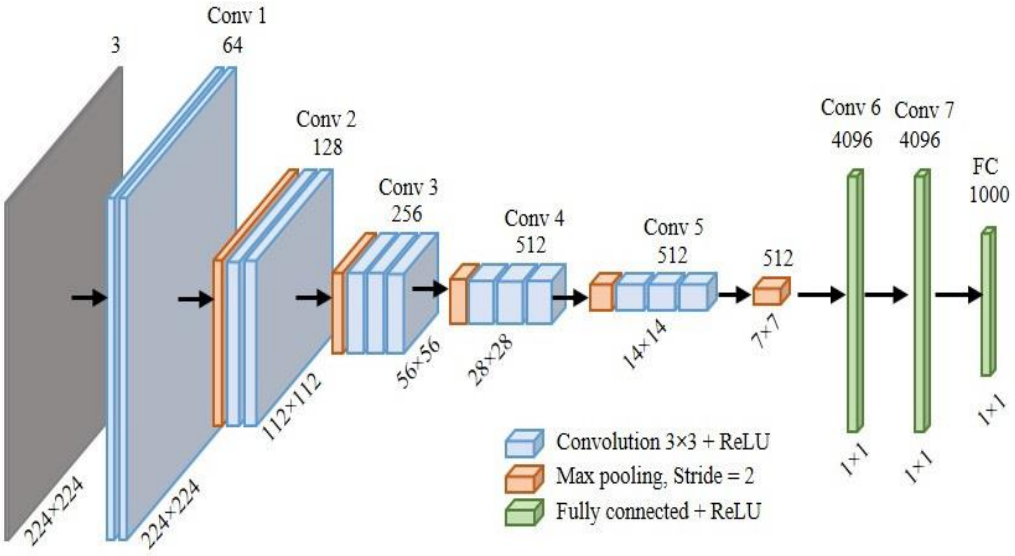


Figure 2.6 VGGNet-16 Architecture [67]

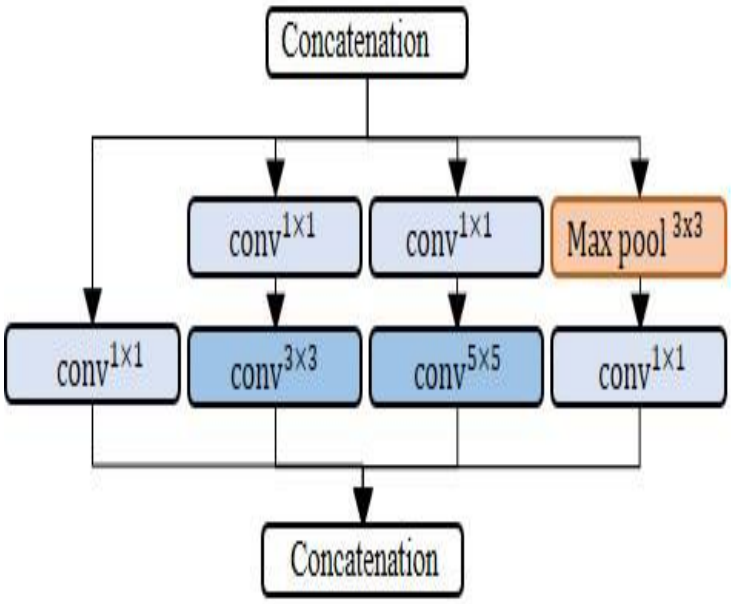


Figure 2.7 Inception Module Architecture [63]

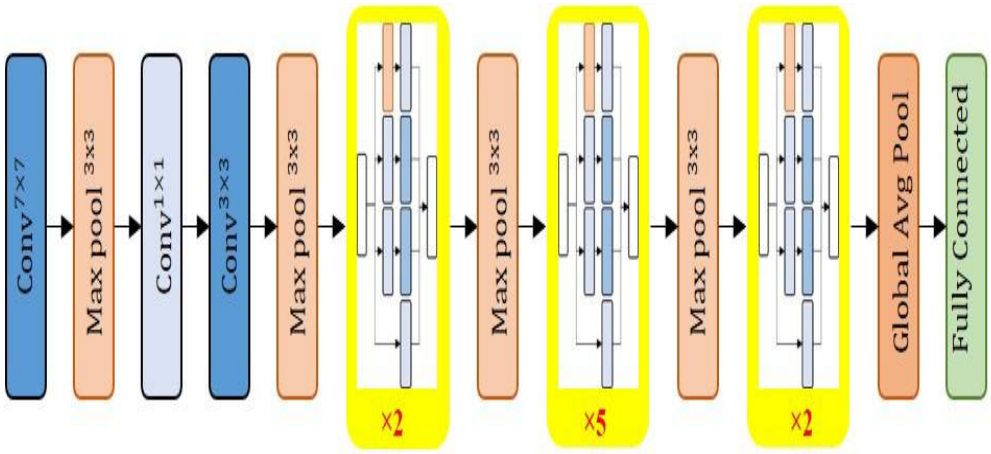


Figure 2.8 22-layer GoogLeNet Architecture [69]

f) ResNet (2016): Since it is more difficult to train deeper neural networks than shallower ones, the development of ResNet marked the start of a new phase in deep neural network training efficiency [70]. In order to facilitate training and optimize the significantly deeper networks, which produced greater accuracy, a residual learning system was developed. Instead of learning unsourced functions, the layers were deliberately reformed to learn residual operations concerning the layer inputs. The introduction of the ResNet Unit (RU), shown in Figure 2.9, was made to address the critical issue [68]. This occurs when adding more layers to a powerful deep model causes the training error to increase. By creating the shortcut interconnection as an identity mapping, ResNet solved this issue. The depth of the residual networks might range from 18, 34, 50, 101, or 152 layers. The most complex ResNet is less complex while being eight times larger than VGGNet. This network demonstrated easier optimization than VGGNet while achieving an increase in object accuracy rate of 28% [71]. In Figure 2.10, the ResNet with a 34-layer residual is displayed. This network has four building blocks, and each has a stack of RU building blocks.

ResNet-34 consists of 18 RU building components in total. Comparing the VGGNet to AlexNet, which has nearly three times as few parameters, involves

much processing. Compared to AlexNet, which has over 60 million parameters, GoogleLeNet's Inception architecture has about 7 million parameters, which is a 9-times reduction. The ability to transport gradients back across all levels in an efficient manner is a worry, though, considering the relatively enormous depth of Google Net's 22 layers. Because shorter networks did so well at this task, we can conclude that the features generated by the middle layers of the network should be very differentiable. This could be used by connecting additional classifiers to the intermediate levels [71]. A deeper system would produce the same classification error as its shallower counterpart using ResNet's shortcut identity mappings [68]. By employing this method, networks containing the Inception module can achieve comparable accuracy while being less expensive [71].

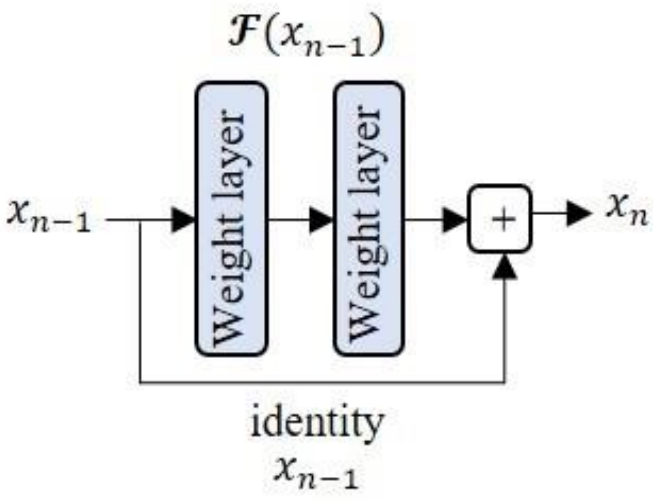


Figure 2.9 A ResNet Unit (RU) [70]

g) DenseNet (2017): The deep learning architecture named DenseNet, or "Densely Connected Convolutional Networks," was developed for image classification and other computer vision problems. "Densely Connected Convolutional Networks," proposed by Huang *et al.* [72], first discussed it in their 2017 publication. DenseNet introduces a special connectivity design among layers to address the issue of disappearing gradients and information flow in deep neural networks. Each layer

in a dense network is directly connected to every layer above it, facilitating information flow and gradients across the network. An architecture with excellent parameter efficiency is produced by this dense connectivity. DenseNet is broken up into a number of dense blocks. The primary innovation within each dense block is that every layer obtains feature maps from all the preceding layers within the same block. Each dense block is made up of several convolutional layers. This encourages feature reuse, enabling the network to learn more condensed and representative features, improving the performance of the network as a whole. The architecture shown in Figure 2.11 has a variety of advantages, such as enhanced gradient flow, feature reuse, a decrease in the number of parameters, and overfitting mitigation. DenseNet models are a popular option in the fields of deep learning and computer vision because of their State-of-the-Art performance on numerous benchmark datasets.

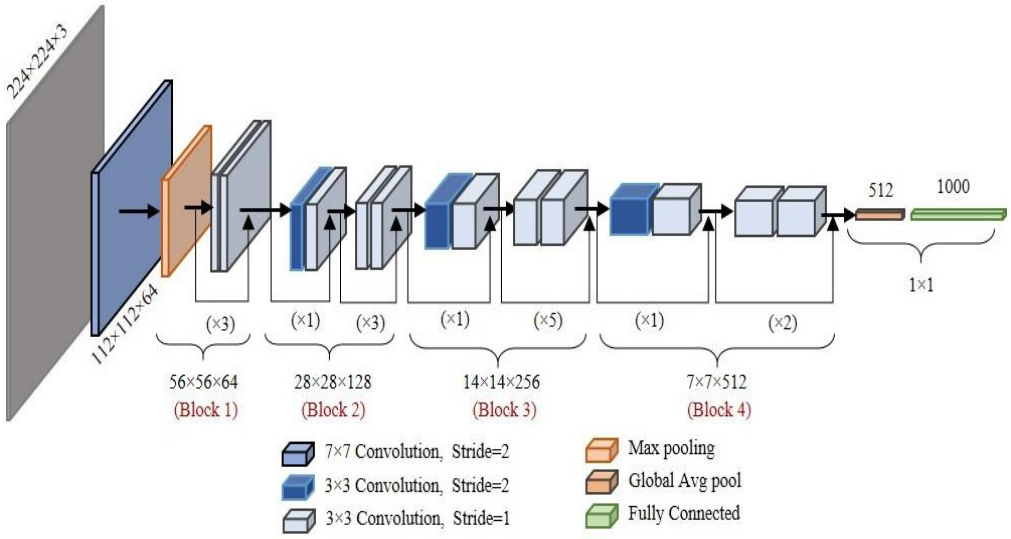


Figure 2.10 A 32-layer ResNet Architecture [70]

The detailed discussion of the pre-trained models is done to show the significance of their use in different applications. The next sub-section provides a detailed overview of the various applications of these pre-trained models. We

utilized these pre-trained models in the proposed and implemented system by incorporating some modifications.

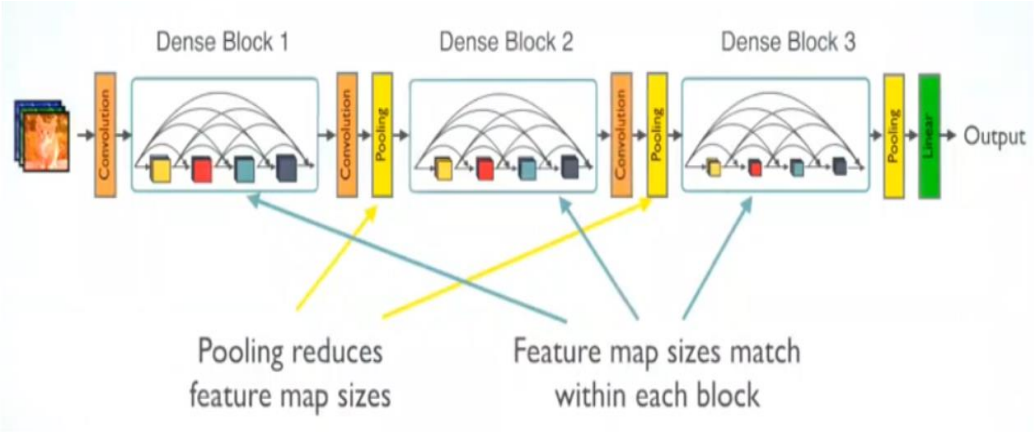


Figure 2.11 The Architecture of DenseNet [73]

2.2.1.2 Advanced Deep CNNs

More sophisticated Deep CNN (DCNN) architectures have adapted the basic L-A-N-V-G-R-D networks for various purposes. Following is a list of advanced DCNNs used in various tasks such as object detection, classification, and segmentation. An illustration of the discussed tasks is given in Figure 2.12.

a) Object Detection

In the field of Computer Vision, object detection refers to the task of simultaneously identifying and precisely locating objects within an image or video frame. Its core objective is not only finding the object but also identifying the correct position of the object, often visualized through bounding boxes encircling the detected objects. It is used in various computer vision tasks such as image and video analysis, surveillance infrastructure, facial recognition, *etc.* Many researchers have proposed various deep learning techniques for object detection. Kuznetsova *et al.* [74] proposed RCNN, a region-based technique using CNN characteristics. The ConvNet structure of AlexNet [61] is swapped out for a 16-layer GoogLeNet

[71] model to build this architecture, resulting in a straightforward and scalable object detection technique. To accurately identify human faces, Taigman *et al.* [31] suggested the nine-layer DeepFace CNN model. DeepID-Net proposed by Ouyang *et al.* [75] tackles a specialized identification problem for distorted objects. To aid in understanding the distortion of object pieces, this framework provides a deformation-limited pooling layer. Although this method is based on Recurrent CNN (RCNN), it is significantly more complicated because the deformation is specified as the visual features at many semantic levels. Dai *et al.* [76] published a subsequent method for modeling transformation matrices. Liu *et al.* [37] proposed the SSD object identification model, which included predictions from several feature maps with different resolutions to recognize objects of various sizes. SSD is much faster than RCNN because it eliminates proposal development and integrates coordinate regression and region classification into a single network. Wang *et al.* [77] recommended the Fully Convolutional Network-based Tracker (FCNT) to address the visual tracking problem. FCNT is a tracker network built on FCN that focuses on high-level features to recognize the semantic class of the object and low-level characteristics to acquire more exclusionary data to more effectively distinguish the same appearance from the background.

b) Classification

Classification is a fundamental task in both the fields of Machine Learning and data analysis. It involves the systematic arrangement of data points or objects into predefined groups or categories, primarily determined by the assessment of their inherent attributes or characteristics. Le *et al.* [78] suggested Deeply-Supervised Nets (DSN) to give a close, integrated look at the hidden layers instead of only supervising the output nodes and sending this information back to earlier levels. They applied the auxiliary classifier to each buried layer, which was regarded as an additional regularizer. However, Szegedy *et al.* [71] had already introduced the importance of the auxiliary classifiers. A highly difficult job of fine-grained

recognition to differentiate among visually very similar things, such as kinds of birds, breeds of dogs, or types of airplanes, was handled by the Deep Convolutional Activation Feature (DeCAF) network presented by Donahue *et al.* [79]. Classification tasks like fine-grained recognition have substantial intra-class and low inter-class variation [79] [80]. Although introducing a residual learning framework with 152 levels made it easier to train deeper networks, the high computing cost of deeper neural networks still makes them difficult to deploy. At that point, the two main issues that need to be handled are the disappearing gradient and model size. Using a feed-forward ResNet technique, Huang *et al.* [72] addressed the gradient vanishing issue by connecting every layer to every other. Their model, DenseNets, also decreased the number of variables. Both ResNet and DenseNet's designs fall under the categories of pre-activation and cross-layer connections. A batch normalization layer follows the convolutional layer in these networks, and the output of one layer can be utilized as the input for numerous following layers. These two well-known deep learning networks were developed to recognize various classes, including the 1000 classes in ImageNet.

c) Pixel Classification

Pixel classification, also known as segmentation, is the process of assigning labels to each pixel of an image that helps to segment the image into regions. Girshick *et al.* [81] suggested an object recognition and semantic segmentation network by fusing several low-level image data points with high-level context. This network uses bottom-up region recommendations in conjunction with CNNs to localize objects and segment them. Another deep neural network, dubbed DeepLab, which enhances the localization of object boundaries, also addresses semantic segmentation [82]. This model incorporates two new elements: the Atrous Spatial Pyramid Pooling (ASPP) module to partition the objects at various scales, and the atrous convolution, a potent tool for controlling the resolution in dense prediction. The Full Resolution Residual Networks (FRNN) model, another DCNN-based

model for semantic segmentation, improves localization accuracy while offering remarkable recognition performance [83].

Most DCNNs have excessive parameters and need millions or even billions of starting point operations. Therefore, deep network designers' primary concerns are storage and computing capacity. One of the primary drivers for reducing the number of these networks' parameters is to increase the effectiveness of their deployment on mobile apps like MobileNet [84] or their training in Internet-scale clusters, which results in lower computing costs and storage requirements. An overview of dimension reduction methods used with deep networks is provided in the next sub-section.

2.2.1.3 Dimensionally Reduced Deep CNNs

The deep networks have dramatically increased accuracy, but there is a significant processing overhead due to the deep networks' enormous number of parameters. Implementing a deep network on hardware systems with constrained processing resources, such as mobile phones, is challenging because of the high storage requirements and computationally expensive floating-point matrix multiplications. Considering ways to lower the memory and computational costs of deep network topologies is crucial. In order to speed up the testing phase of the large-scale training network, Denton *et al.* [85] devised a linear compression algorithm. This method cuts the test time two-fold by taking advantage of CNNs' linear nature.

The RCNN object detection network requires a lot of computing power. Two improved versions of this network are being made to make it work better. The first one, called fast-RCNN, finds the Region of Interest (RoI) [86], and the second one, also called fast-RCNN, builds the Region Proposal Networks (RPNs) on top of the RCNN convolutional feature mappings [87]. FitNets is a framework that Adriana *et al*. [88] created to condense a wide, deep network with many parameters into a

deeper, thinner network with fewer parameters. The compressed network is trained using intermediate-level cues from the larger network. FitNets has shown that deeper and narrower networks can generalize and operate more quickly than wider ones. Han *et al.* [89] proposed a parameter reduction technique to reduce computational time and memory consumption in CNNs by deleting extraneous links in the first round of learning and then fine-tuning only the critical connections in the second. This method maintained the accuracy of AlexNet, which had nine times fewer parameters, and VGG-16, which had 13 times fewer parameters. ThiNet [90] used filter level pruning as another optimization strategy by removing the less significant filters. Instead of using the statistics from the current layer, ThiNet prunes the filters depending on the statistics from the next layer. It reduced the VGGNet-16 model's size by a factor of 16 while just slightly decreasing accuracy. SqueezeNet, proposed by Iandola *et al*. [91], is a compressed version of AlexNet that keeps accuracy while having 50 times fewer parameters. The re-module, which this technique introduced, has two different sorts of layers: the compress convolution and the expansion. In a different architecture known as Deep Fried Convnets, the fully connected layers are re-parametrized using an adaptive Fastfood transform algorithm because they contain high and over 90% of the CNN parameters [92].

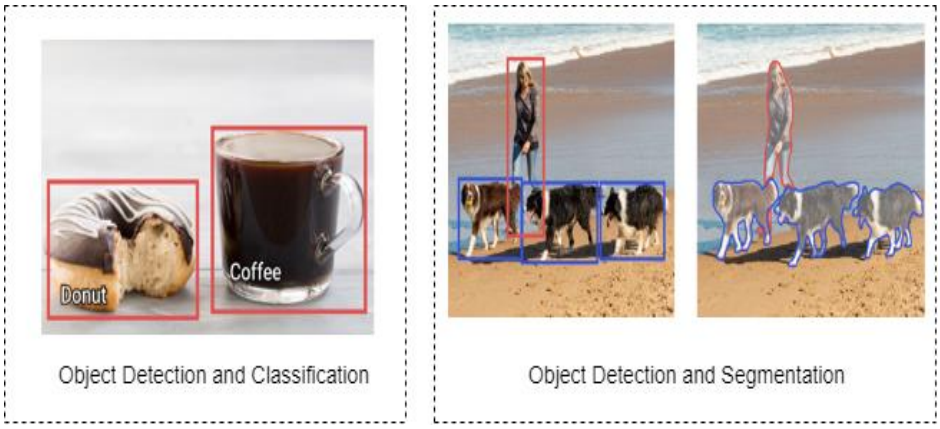


Figure 2.12 Applications of Advanced Deep CNNs

2.2.1.4 Existing State-of-the-Art Approaches for Face Recognition using Deep Learning

In recent times, Deep Convolutional Neural Networks (CNNs) have demonstrated remarkable achievements across a range of computer vision tasks, particularly in the realm of object detection. These deep CNN models have proven their ability to effectively capture diverse variations present in large datasets and learn discriminative nonlinear feature representations. Consequently, they have emerged as powerful tools for face recognition (FR) applications by directly learning effective feature representations from face images [26] [93] [94]. For instance, DeepID, DeepID2, and DeepID2+ were introduced in [27] [95] [96] to acquire a set of high-level discriminative feature representations. DeepID [95] is trained by employing a collection of small CNNs and achieves an impressive recognition accuracy of 97.45%. Each CNN is individually fed with specific facial image regions such as the eyes, nose, and mouth, and the learned features are combined to form a powerful model. Expanding on this research, subsequent studies [27] [96] amplified the feature dimension of the last hidden layer and leveraged the hierarchical and non-linear characteristics of the convolutional layers. This approach facilitated the acquisition of hierarchical and nonlinear feature representations, intended to better differentiate between different individuals by extracting unique traits from each identity while minimizing variations within the same individual. In contrast to the DeepID series, Microsoft DeepFace [31] integrates precise facial alignment to extract a resilient facial representation through a 9-layer deep CNN. Developed by Facebook, DeepFace [31] achieved a remarkable recognition accuracy of 97.35% in face recognition, comparable to human-level performance. The model consists of over 120 million parameters and is trained on a dataset of 4.4 million images belonging to 4000 identities. Training such a model requires several days and highly computational systems. In an alternative investigation [97], the emphasis shifts from individual faces to the

concurrent extraction of high-level facial similarity attributes from face pairs. This strategy entails employing numerous deep CNNs meticulously tailored for face verification tasks. Likewise, in the context of Single Image of Person (SIP) challenges, recent studies [98] [99] [100] adopted a loss function based on triplets to acquire robust facial embedding. The goal of this loss function is to distinguish between pairs that are positive and match the same facial regions of interest (ROIs) and pairs that are negative and match distinct face ROIs. An alternative approach presented in [101] involves optimizing the triplet loss to acquire a robust facial representation. This optimization is achieved through a streamlined and rapid Cross-Correlation Matching CNN (CCM-CNN). Autoencoder neural networks offer another avenue to extract deterministic nonlinear feature maps that are resilient to various factors affecting facial images, such as lighting, expression, and poses [102] [103]. The autoencoder architecture comprises encoder and decoder components, wherein the encoder converts input data into latent nodes, while the decoder reconstitutes these latent nodes back into the initial input data domain. The goal is to minimize the reconstruction error [102]. Inspired by [104], several autoencoder networks have been suggested to address the mentioned abnormalities in facial images [102] [105] [106]. These networks treat faces with different variations as noisy images and aim to address lighting, pose, and other factors. The authors in [106], describes an investigation using a CNN based on facial components to convert faces with different lighting and poses into frontal-view faces. It accomplishes this by utilizing pose-invariant features from the final hidden layer as facial representations. Several deep architectures have also been introduced, and these architectures employ multitasking learning to rotate faces with arbitrary poses and varying lighting conditions into target pose faces [107] [108]. Furthermore, the comprehensive architecture presented in [109] encodes a desired attribute, merging it with the input image to generate target images that retain similarity to the input image while integrating changes in visual attributes (*e.g.*, lighting, facial appearance, or pose) while keeping other aspects of the face

unaltered. In addition, a supervised deep architecture known as FlowNet [110] tackles the estimation of optical flow by precisely predicting flows through the correlation of feature vectors derived from pairs of images located at different positions. In the context of Single Image of Person (SIP) scenarios, a deep supervised autoencoder proposed in [101] maps non-frontal faces with various complicating circumstances to the canonical face, a frontal face with neutral expression and normal lighting, of the same person in order to learn a robust face representation. However, due to their computational intricacies and the distinctions between static images and video frames, these methods may not be optimally suited for S2V FR tasks. To overcome these challenges and address the limitations of domain matching, researchers proposed a solution called the supervised autoencoder-based Canonical Face Representation CNN (CFR-CNN) in [102]. This methodology forms the foundational framework for a S2V FR system that centers around domain alignment by the reconstruction of frontal faces from specific video Regions of Interest (ROIs). To facilitate the matching of input probes, a separate, fully connected network was trained as a classifier. The development of an accurate depth model necessitates the simultaneous consideration of both static images and videos during network training and optimization. Furthermore, to address the variations inherent in the Single Image of Person (SIP) context, a supervised autoencoder was introduced. This autoencoder maps diverse facial variations to a canonical representation of a single individual's face [102]. Advances in frontal view synthesis and pose-invariant representation acquisition through an adversarial process have led to the development of Generative Adversarial Networks (GANs) [111] [112]. For example, a two-path GAN simultaneously manages the overall facial structure and the transformation of local intricacies. However, these methodologies require landmark detection and may not comprehensively accommodate variations such as blurring and scale changes (due to subjects' distance from surveillance cameras), thereby making them less suited for video-oriented face recognition applications.

Other face recognition algorithms rooted in deep learning, like Deep Face Recognition [97], have demonstrated impressive achievements. For instance, they achieved a recognition accuracy of 98.95% on the Labeled Faces in the Wild (LFW) dataset and 97.3% on the YouTube Faces (YTF) dataset. In [113], a robust CNN was trained using a combination of softmax loss and center loss to extract deep features that improve both between-class variability and within-class compactness, which are crucial for accurate face recognition. The resulting model achieved remarkable recognition accuracy rates of 99.28% on the LFW dataset and 94.9% on the YTF dataset.

Training a DCNN from scratch requires a substantial amount of training data, making it challenging to achieve proper model convergence, especially in scenarios where privacy is a primary concern. To address data scarcity and overfitting issues, a technique called data augmentation is employed, entails generating new data by applying small adjustments to the current dataset, including flips, rotations, mirroring, translations, *etc.* [114]. Data augmentation helps mitigate the shortage of data and improves the generalization capability of the model. Retraining a CNN from another network with pre-trained settings is a promising additional strategy to address data scarcity [55]. Modern image classification networks that have been trained on millions of photos from a particular domain are called pre-trained CNNs. They can be used for many domains of interest after undergoing several weeks of training across several servers. This approach has proven to be highly valuable for researchers facing resource constraints, as it allows them to leverage the knowledge and features extracted by these large pre-trained models for their specific area of interest. Researchers can achieve optimal performance for their application based on the available data, providing a practical and effective solution by fine-tuning an existing model.

2.2.2 Techniques to Optimize Deep Learning Models

To enhance the efficiency of deep learning-based algorithms, the subsequent approaches can be implemented to mitigate the model's training time [115].

a) Backpropagation: Utilizing backpropagation techniques is an effective way to compute the gradient function during each iteration. This approach within deep learning employs gradient-based methods to address optimization challenges [116].

b) Stochastic Gradient Descent (SGD): It efficiently locates the optimal minimum through the utilization of convex functions by disregarding local minima. The determination of the optimal minimum across diverse trajectories is influenced by parameters such as step size, learning rate, and activation function values [117]. The mathematical equation for SGD to update the model's parameter is given in equation (2.1).

$$\theta_{i+1} = \theta_i - \eta \nabla \mathrm{J}\,(\theta_i;\, x^{(i)}, y^{(i)}) \tag{2.1}$$

Here, θ_i represents the model's parameters at iteration$i$, η is the learning rate, and $\nabla \mathrm{J}\,(\theta_i;\, x^{(i)}, y^{(i)})$ is the gradient of the loss function J.

c) Learning Rate Decay: Modifying the learning rate leads to a reduction in the training time of gradient descent algorithms while concurrently enhancing the model's overall performance. This approach finds extensive application due to its capacity to effect substantial changes during the initial training stages, subsequently gradually diminishing the learning rate. Moreover, this technique enables fine-tuning of weights in subsequent iterations and is mathematically represented by equation (2.2) [118].

$$k = i \times \frac{1}{1 + d \times \frac{k}{step\ size}} \tag{2.2}$$

Here, k is the learning rate, i is the initial learning rate at the beginning of training the mode, d is the decay rate at which the learning rate decreases, and step size is the number of epochs before each decay.

d) Max-Pooling: Across non-overlapping segments of the input layer, a pre-configured filter is employed to extract maximum values and generate the resulting output. The application of the max-pooling technique also brings about a reduction in computational expenses associated with learning multiple parameters [66] [119] and is mathematically represented as given in equation (2.3).

$$P = O_{max}^{n,n}(F) \tag{2.3}$$

Here, F is the input feature map of size $n \times n$ obtained from the previous convolutional layer.

e) Dropout: Tailored for the challenge of neural network overfitting, the dropout technique employs a strategy of randomly omitting units and their connections throughout the training phase. For a single neuron in a neural network layer, the output ρ after dropout is applied can be calculated using equation (2.4). This technique serves as an improved regularization approach, effectively curbing overfitting within neural networks and ameliorating generalization error [120]. In the realm of deep learning, this method garners superior results for supervised learning tasks [121].

$$\rho = \frac{1}{1-p}.x.d \tag{2.4}$$

Here, x is the output of the neuron before applying dropout, d is the binary dropout mask obtained from the Bernoulli distribution with probability p.

f) Batch Normalization: Batch normalization reduces covariate shift, which increases the learning rate of deep neural networks. During the training process, for each small batch, this method normalizes the input layer as the weights are adjusted. Enhanced network stability is achieved through the normalization of output from

the final activation layer. Furthermore, batch normalization methodologies contribute to improved learning rates and a reduction in the required training epochs [122].

g) Transfer Learning: In transfer learning, a model initially trained for a particular task is adapted to undergo training for a comparable task. The knowledge acquired from addressing one challenge can be efficiently utilized to tackle another related issue. This process expedites advancements and enhances performance when addressing the second related task [24].

h) Ensemble Learning: In Machine Learning, ensemble techniques combine several models or classifiers to generate an ideal model that produces precise predictions for the intended result. Ensemble learning is employed to enhance recognition accuracy by averaging the weights of multiple deep learning models [123].

Based upon the aforementioned techniques for optimizing deep learning models, a comparison of their respective advantages and disadvantages is presented in Table 2.2.

Table 2.2 Pros and Cons of Optimization Techniques

S. No.	Technique	Description	Pros	Cons
1.	Back Propagation	Used in the optimization problems	Used to calculate the gradient	Susceptible to the effects of noisy data
2.	Stochastic Gradient Descent	Locate optimal minima in optimization problems	Prevents getting into local minima	Convergence time is large, demanding substantial computational resources
3.	Learning Rate Decay	Reduce the learning rate gradually	Enhancing the performance of the model helps reduce training time	Demanding significant computational resources

4.	Max-pooling	Downsampling technique for feature extraction	Reduces dimensionality and computational overhead	Considers only the maximum value of the region of an image, which may lead to an unacceptable result
5.	Dropout	Random deactivation of neurons during training	Prevents overfitting	Increases the training time required for the model to converge
6.	Batch Normalization	Nomalizing the activations of a layer within a mini-batch of data	Reduction in covariate shift, stable and faster convergence, and improved generalization	Increases implementation complexity and slows down the training of the model
7.	Transfer learning	Utilization of the knowledge of first model to resolve another problem	Handles data scarcity, improves performance of the model, and prevents overfitting	Limited flexibility (i.e., can work with similar types of problems)
8.	Ensemble learning	Prediction of each model is averaged to get the final prediction	Improves recognition accuracy	Computational overhead during training

2.2.3 Deep Learning Framework

Deep learning frameworks facilitate the expedited design of neural networks by obviating the need for delving into the intricacies of underlying algorithms. Typically, each framework is tailored to address specific problem statements [124]. The subsequent table, Table 2.3, provides a succinct overview of several deep learning frameworks.

a) Fast.ai: Built on PyTorch, the user-friendly, open-source fast.ai deep learning library places a strong emphasis on simplicity and effectiveness. It offers a high-level API that makes data augmentation, transfer learning, and model training simpler. Fastai promotes data comprehension and preprocessing with a focus on

organized deep learning. For quicker convergence, it incorporates cutting-edge methods like learning rate annealing and progressive resizing. Through the use of visualization tools, the library facilitates model interpretation. It encourages community cooperation, provides courses, and has a wealth of documentation. Due to its user-friendly design, it is particularly beneficial for beginners exploring the deep learning discipline [125].

b) PyTorch: It serves the dual purpose of constructing deep neural networks and performing tensor computations. As a Python-based package, PyTorch offers tensor computation capabilities and a framework for generating computational graphs.

c) Keras: Built atop TensorFlow, the Keras Application Programming Interface (API) is coded in Python. This interface facilitates rapid experimentation and extends support to CNN as well as Recurrent Neural Networks (RNN). It provides the same deep learning model deployment capabilities on CPUs and GPUs as TensorFlow.

d) TensorFlow: TensorFlow offers compatibility with a range of modern programming languages, including C++, Python, and R. This framework enables the seamless deployment of deep learning models on both Central Processing Units (CPUs) and Graphics Processing Units (GPUs), and was developed by Google Brain.

e) Deeplearning4j: Implemented in Java, Deeplearning4j exhibits superior efficiency compared to Python. Utilizing the ND4J tensor library, it empowers the manipulation of multi-dimensional arrays or tensors. This framework is compatible with both CPUs and GPUs. Deeplearning4j seamlessly handles diverse data formats, including images, CSV, and plaintext.

f) Caffe: Caffe, developed by Yangqing Jia, is an open-source framework. What distinguishes Caffe from other frameworks is its rapid processing speed and

proficiency in learning features from images. Pre-trained models are made available through the Caffe Model Zoo framework, which makes the solution of various problems easy.

Table 2.3 Comparison of Deep Learning Framework

S. No	Deep Learning Framework	Release Year	Written in Language	CUDA Supported	Pre-trained Model
1.	Fast.ai	2018	Python	Y	Y
2.	Pytorch	2016	C, Python	Y	Y
3.	Keras	2015	Python	Y	Y
4.	TensorFlow	2015	C++, Python	Y	Y
5.	Deeplearning4j	2014	C++, JAVA	Y	Y
6.	Caffe	2013	C++	Y	Y

(*Y=Yes)

2.3 Transfer Learning or Domain Adaptation-based Techniques for Face Recognition

Transfer learning can be employed to enhance classification performance by transferring knowledge from a domain that has ample unlabeled data to a domain with limited labeled data. This approach is useful when there are differences in data distribution or feature spaces between the training and test datasets. In scenarios where collecting and labeling new data can be costly and time-consuming, transfer learning offers an attractive strategy compared to traditional Machine Learning approaches [126]. Transfer learning involves utilizing a pre-trained model as a foundation for a new machine learning task, leveraging the knowledge gained during the initial training to improve learning and performance on a related task. This approach is particularly beneficial when the new task has concise annotations or the data distribution differs from the original task. By reusing the pre-trained model, transfer learning saves time and computational resources while enhancing accuracy and generalization [23] [127]. Essentially, the investigation of various domains, tasks, and distributions between the training and testing stages is made possible by transfer learning. In transfer learning, the importance of the target task

takes precedence over the source task since the model is fine-tuned for the target task [127]. Transfer learning can be classified into two settings: (1) inductive and (2) transductive, as defined by equations (2.5) and (2.6). Inductive transfer learning uses distinct target and source tasks, with some labeled data available in the target domain. Conversely, transductive transfer learning addresses distinct source and destination domains while preserving the same task. In transductive transfer learning, labeled data from the source domain and unlabeled data from the target domain must be used to adjust the learning function, as is the case in Domain Adaptation (DA) [127]. The scarcity of datasets, especially in scenarios where privacy is a significant concern, has driven the application of transfer learning techniques by researchers [24]. For the Single Image of Person (SIP) problem, a discriminative transfer learning approach has been proposed. In this approach, a generic training set (source domain) is used to learn a feature projection that is then transferred to a single-sample gallery set (target domain) through discriminant analysis. This approach aims to minimize the differences between the source and target domains and incorporates sparsity regularization to enhance robustness against outliers and noise [128]. Alhanaee *et al.* [129] used deep transfer learning and pre-trained models to figure out the accuracy of their dataset's detection in the context of face recognition-based intelligent attendance systems. However, these State-of-the-Art approaches have limitations in that they were evaluated on fundamental datasets that contain only a small number of uncontrolled factors.

Inductive transfer learning

$$if\ S_D \neq T_D\ or\ L_s \neq L_t, \tag{2.5}$$

It improves the learning of $f(.)$ in T_D by applying the knowledge in $S_D\ and\ L_s$, where $L_s \neq L_t$

Transductive transfer learning

$$if\ S_D \neq T_D\ or\ L_s = L_t, \tag{2.6}$$

It improves the learning of $f(.)$ in T_D by applying the knowledge in S_D and L_s, where $L_s = L_t$

Here, S_D is the source domain, and L_s is the learning task of the source domain, T_D represents the target domain, and L_t signifies the learning task within the target domain, the functions $f(.)$ serves as the predictive function.

2.4 Ensemble Learning-based Techniques for Face Recognition

Ensemble techniques within the realm of machine learning involve the amalgamation of multiple models or classifiers to create an optimal composite model that delivers accurate predictions for the intended outcomes. The main idea behind ensemble models is to use the best parts of different learning algorithms at the same time. Compared to single models, this lets ensemble models make better predictions. The precision of a classifier is inherently linked to the quality of features extracted or learned from input data, such as images. Nevertheless, through the fusion of numerous classifiers and the amalgamation of their outcomes, further refinement of accuracy becomes feasible. Ensemble classification systems have garnered considerable attention across various domains, encompassing fields like face recognition [130], geospatial land classification [131], video-based face recognition systems [132], medical image segmentation [133], and wind power forecasting [134]. These models exhibit heightened accuracy by effectively mitigating overfitting concerns and curtailing bias and variance errors in comparison to individual classifiers. The value of ensemble models is underscored by their triumphant application in renowned machine learning competitions, exemplified by the likes of the Netflix challenge [135], the Knowledge Discovery in Databases (KDD) Cup 2009, and Kaggle, where ensemble-based models clinched top-ranking accuracy scores.

The performance of the classifier is greatly increased by the introduction of multi-classifier-based systems, in which the output of separate base classifiers is

combined [100] [132] [136]. Pattern recognition tasks with sparse and uneven training data are especially well-suited for ensemble approaches. Ensemble techniques' main concept is to create a variety of classifiers from the original data and combine them to get predictions that are better than those of any one basic classifier [137] [138]. Numerous studies have demonstrated that ensemble methods bolster the resilience and accuracy of classification systems [137] [138]. Key considerations in ensemble-based systems include the accuracy and diversity of the classifiers within the ensemble [139] [140]. While accurate classifiers are desirable, it is also crucial for the classifiers to be distinct from each other. Selecting the best classifier from the ensemble should not solely rely on training data accuracy. It is essential to incorporate diversity among the classifiers in the ensemble to ensure effectiveness. This can be achieved through various approaches, as outlined below [123]:

- Using same classification algorithm with different instantiation or different hyper-parameter settings.
- Using different classification algorithms for ensemble system.
- Using different feature sets:
    - Random selection
    - Feature selection
- Using different training sets:
    - Bagging
    - Boosting
    - Stacking

a) Bagging or Bootstrap Aggregating: An ensemble technique widely acknowledged in the field involves using a non-hybrid classifier, applying the same classification algorithm with various instantiations or hyperparameter settings to create an ensemble model. Bagging, another name for bootstrap aggregation, is an early ensemble-based method that is simple to understand. It operates by training

multiple models using subsets of randomly chosen datasets from the original training set with replacement. A majority decision among the individual classifiers determines the ensemble's prediction. Figure 2.13 illustrates the process flow of the bagging technique. There are several variations of this algorithm aimed at enhancing the model's performance.

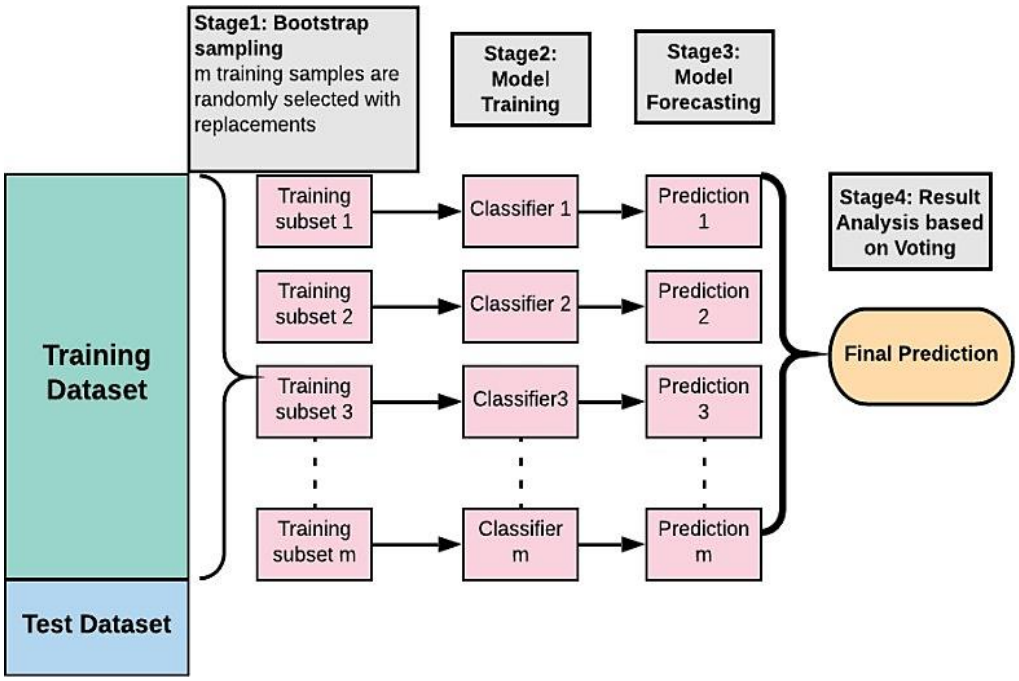


Figure 2.13 Parallel Execution of Bagging Process

b) Boosting: A variant of the bagging technique known as boosting is employed to enhance the classification model by sequentially transforming weak learners into strong learners, with each learner attempting to correct its predecessor. The fundamental distinction between bagging and boosting lies in their training approaches. In boosting, the architecture of the current model is dependent on the performance of earlier classifiers, but in bagging, each model is built independently during a parallel training period. Boosting is a sequential procedure in which the data is first given similar weights, and these weights are then redistributed following each training phase. This redistribution allows subsequent learners to

place greater emphasis on misclassified cases, which are now assigned higher weights. Figure 2.14 illustrates this process.

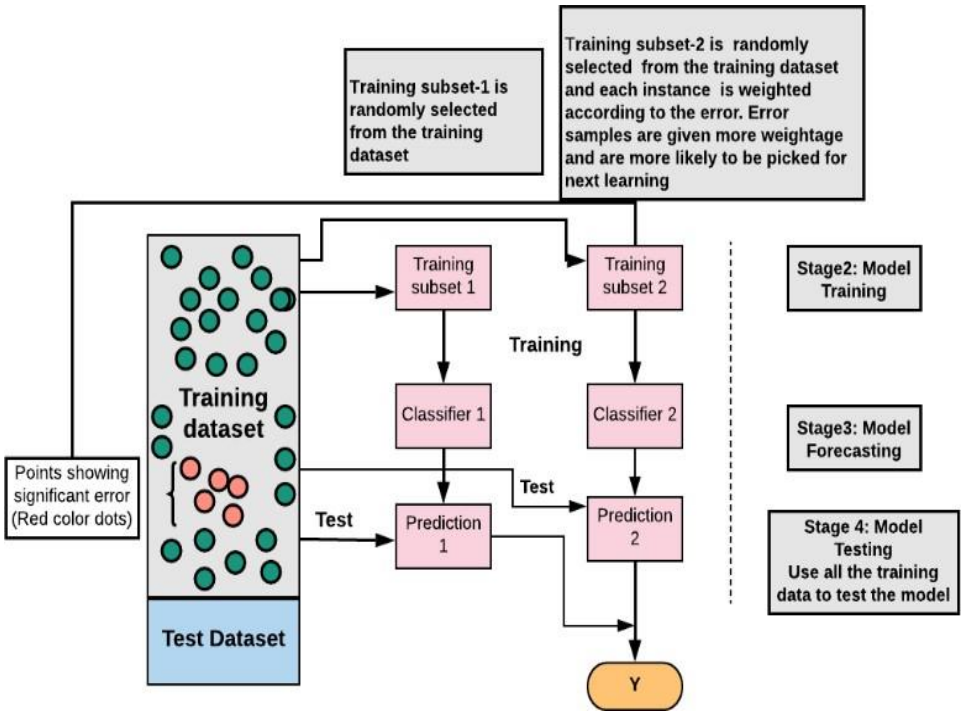


Figure 2.14 The Flow of the Execution of the Boosting Process

c) Stacked Ensembles: Stacking is a multi-layer learning technique in which base learners make up the first layer while lower-level meta-learners then use the base learners' outputs to figure out the optimal set of first-level models. The concept of the Super Learner (SL) was initially introduced in 1992 [141], and its implementation with enhanced performance was demonstrated in 2007 [142], highlighting the effectiveness of stacked ensembles in creating an optimal learning model. Random Forest (RF) is a well-known machine learning algorithm that uses the bagging technique. As Figure 2.15 illustrates, RF combines a collection of weak learners, such decision trees, to build a single powerful learner.

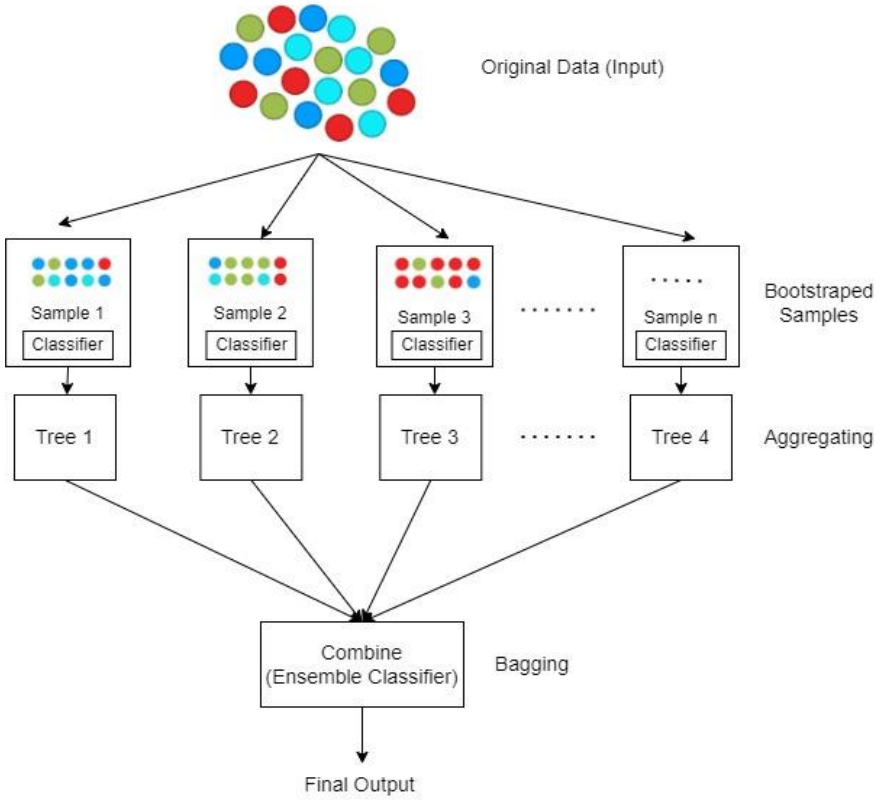


Figure 2.15 The Flow of the Execution of the Stacking Process

2.4.1 Existing State-of-the-Art Approaches for Face Recognition Using Ensemble Learning

A Convolutional Neural Networks (CNN)-based framework proposed by Ding *et al.* [132] addressed the challenges in video-based facial recognition. They introduced the Trunk-Branch Ensemble CNN (TBE-CNN) model to handle pose and occlusion variations. The TBE-CNN was trained using the Mean Distance Regularized Triplet Loss (MDR-TL) function. The proposed method was evaluated on multiple video datasets, including COX Face, PaSC, and YouTube Faces. Impressive recognition accuracies were achieved, such as approximately 95% on the YouTube Faces dataset, 96% on the PaSC dataset, and 99.33% accuracy for V2V, 98.96% for V2S, and 95.74% for S2V on the COX dataset. Their approach secured first place in the BTAS 2016 Video Person Recognition Evaluation. The

proposed approach effectively addressed challenges such as blur, partial occlusion, and pose variations. Tang *et al.* [136] proposed an ensemble model combining CNN and Local Binary Pattern (LBP) for face recognition. LBP was used to extract texture-related features from the face, and ten convolutional neural networks with five different network structures were employed to extract features and obtain classification results in the fully connected layer. The face recognition result was obtained using parallel ensemble learning with majority voting. In another study [97], an ensemble of CNN models was trained using holistic facial images and multiple overlapping and non-overlapping visual fields to handle pose and partial occlusion variations. Fusion of these models was achieved through feature chaining to construct over-complete and compact representations. However, specialized CNN models like TBE-CNN [132] and HaarNet [100] can enhance robustness to facial appearance variations at the expense of increased computational complexity. In these models, complicated and asymmetric face traits are captured by branch networks, and the root network captures the overall facial look (holistic representation). For example, TBE-CNN uses face landmarks, and HaarNet uses three branching networks based on Haar-like features. However, these complex CNN models may not be suitable for real-time face recognition applications [143]. Therefore, there is a need for a simple ensemble model that can provide high accuracy with fewer computations.

2.5 Challenging Areas of Face Recognition

Face recognition from images and videos presents significant challenges, and extensive research has been conducted to achieve high precision. However, satisfactory results are yet to be attained due to various factors that affect the performance of these systems. These factors include occlusion, low resolution, noise, illumination, pose variation, expressions, aging, and plastic surgery [4] [144]. These can be classified into two main groups: intrinsic and extrinsic factors [4]. Intrinsic factors are tied to the inherent attributes of the human face, including

aging, facial expressions, and plastic surgery, directly influencing the system. Conversely, extrinsic factors entail alterations in facial appearance like occlusion, low resolution, noise, illumination, and pose variation, as depicted in Figure 2.16.

a) Occlusion: Partial occlusion emerges as a notable obstacle in the realm of face recognition endeavors. The concealment of specific facial features impedes the precise identification of individuals. For instance, eyeglasses or sunglasses can obscure the eyes; earrings or hair might veil the ears; scarves could shroud a substantial portion of the face; and facial hair like moustaches and beards might obscure significant facial attributes, as portrayed in Figure 2.16 (a). These factors have a detrimental effect on the performance of face recognition systems. Researchers have been investigating various approaches to address these challenges [145] [146].

b) Low Resolution: Figure 2.16 (b) illustrates that pictures captured from surveillance video cameras often contain small faces, resulting in low resolution. Comparing a low-resolution query image with a high-resolution gallery image poses a significant challenge. The limited data in a low-resolution image leads to the loss of many important details, which can significantly degrade recognition accuracy. Researchers have been exploring various approaches to address this issue [145] [147].

c) Noise: Digital images are susceptible to different types of noise, which can result in poor accuracy in detection and recognition tasks. The introduction of noise into images can occur through various means, depending on how the image is created. Pre-processing plays a crucial role in the overall face detection and recognition system [148]. Figure 2.16 (c) visually depicts the original image along with the presence of salt and pepper noise.

d) Illumination: The variations in illumination can have a significant negative impact on the performance of face recognition systems. Various factors, such as

background light, shadow, brightness, and contrast, can contribute to these variations. Figure 2.16 (d) illustrates images captured under different lighting conditions. Several approaches to addressing illumination-related challenges are discussed in [149] [150] [151].

e) Pose Variation: Pose variation poses a significant challenge for face recognition systems. Matching a profile face with a frontal face in the gallery requires frontal face reconstruction [12]. This reconstruction is necessary because dataset images typically contain frontal views, and matching non-frontal profile faces can lead to inaccurate results. Researchers have proposed various approaches to convert non-frontal faces to frontal faces, which can improve recognition accuracy [12] [51]. The detrimental impact of pose variation on algorithm performance is extensively discussed in the proposed approaches [45] [152] [153]. Figure 2.16 (e) illustrates the different pose distributions of an individual.

f) Expressions: Facial expressions play a crucial role in expressing our emotions, as depicted in Figure 2.16 (f). They can alter the facial geometry, and even a slight variation can introduce ambiguity for face recognition systems. Muscle contractions that occur quickly lead to changes in facial features like the mouth, cheeks, and eyebrows, which are all part of facial expressions. Ongoing research focuses on incorporating facial expressions into face recognition methods [40] [149].

g) Aging: Aging is a natural factor that significantly impacts face recognition systems, often posing challenges for algorithms. The face comprises various components, including skin tissues, facial muscles, and bones. When muscles contract, they cause distortions in facial features. However, aging brings about substantial changes in facial appearance, such as changes in facial texture (*e.g.*, wrinkles) and face shape over time. Face recognition systems should be capable of

addressing these changes [154] [155]. Figure 2.16 (g) illustrates the different textures of the faces of the same individual at various ages.

h) Plastic Surgery: Plastic surgery is another significant factor that can impact the accuracy of face recognition. Incidents have occurred where individuals have undergone plastic surgery due to accidents, resulting in their faces becoming unrecognizable to existing face recognition systems. This factor is particularly relevant in cases where criminals attempt to alter their identities through plastic surgery. Therefore, as highlighted in [156], there is a need for an identification system capable of recognizing faces even after reconstructive surgery. The impact of plastic surgery on facial appearance is depicted in Figure 2.16 (h).

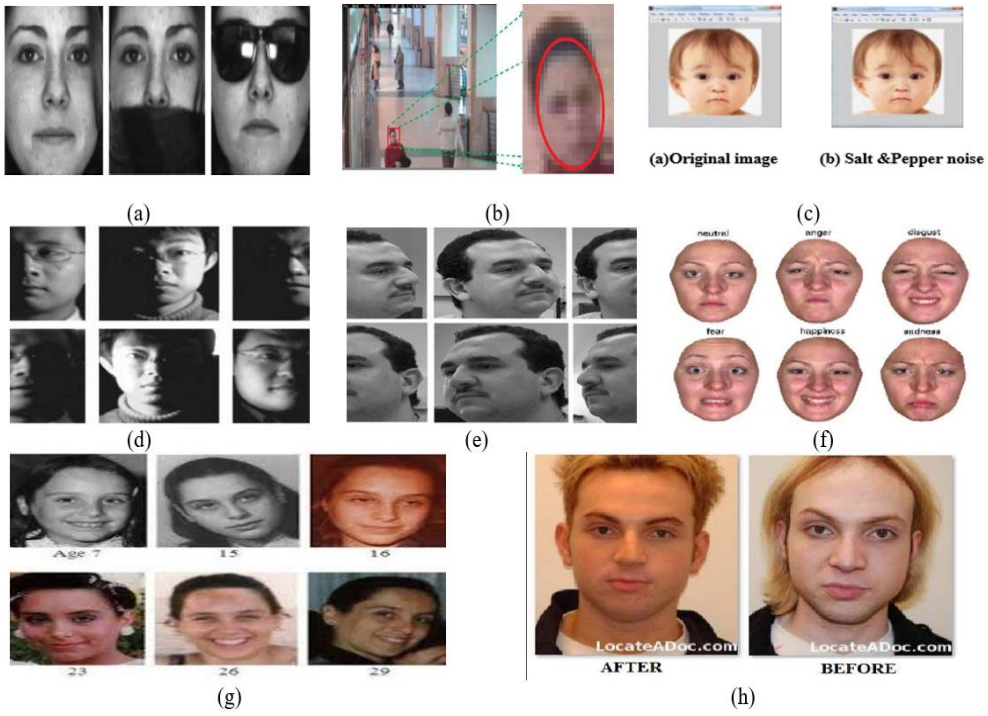


Figure 2.16 Factors Affecting Facial Recognition Accuracy

2.6 State-of-the-Art Datasets for Face Recognition

In the course of the last three decades, numerous face datasets have been created, reflecting a clear trend towards larger scales, diverse sources, and real-

world unconstrained conditions. As simpler datasets such as LFW reached performance saturation, the development of increasingly complex datasets became essential to facilitate further research in face recognition [157]. It is fair to say that the evolution of face datasets played a significant role in shaping the direction of face recognition research. In this section, we present Table 2.4, which includes freely available training datasets, and Table 2.5, which lists the testing datasets specifically designed for deep face recognition tasks.

Table 2.4 Publicly Available Training Datasets for Face Recognition

S. No.	Datasets	Publication Year	No. of Images	No. of Classes	No. of Images per Class (Min/Average/Max)
1.	MillionCelebs [158]	2020	18.8 million	6,36,000	29.5
2.	MS-Celeb-1M (Challenge 3) [159]	2018	4 million	80,000	NR
3.	IMDB-Face [160]	2018	1.7 million	59,000	28.8
4.	VGGFace2 [28]	2017	3.31 million	9,131	87/362.6/843
5.	UMDFaces-Videos [161]	2017	22,075	3,107	NR
6.	MS-Celeb-1M (Challenge 1) [162]	2016	10 million	100,000	100
7.	MS-Celeb-1M (Challenge 2) [162]	2016	1.5 million	20,000	1/NR/100
8.	MegaFace [163]	2016	4.7 million	6,72,057	3/7/2469
9.	VGGFace [98]	2015	2.6 million	2,622	1,000
10.	CASIA WebFace [164]	2014	4,94,414	10,575	2/46.8/804

(*NR is Not Reported).

Table 2.5 Publicly Available Testing Datasets for Face Recognition

S. No.	Datasets	Publication Year	No. of Images	No. of Classes	No. of Images per Class (Min/Average/Max)

1.	IJB-C [165]	2018	3,13,000 images 11,779 videos	3531	42.1
2.	RFW [166]	2018	40,607	11,429	3.6
3.	IJB-B [167]	2017	11,754 images 7,011 videos	1.845	36.2
4.	CPLFW [42]	2017	11,652	3,968	2/2.9/3
5.	CALFW [168]	2017	12,174	4,025	2/3/4
6.	CFP [169]	2016	7,000	500	14
7.	UMDFaces [170]	2016	3,67,920	8,501	43.3
8.	IJB-A [171]	2015	25,809	500	11.4
9.	COX-S2V [52]	2015	NR	1,000	1 image, 4 video clips
10.	PaSC [172]	2013	2,802	265	NR
11.	YTF [44]	2011	3,425	1,595	48/181.3/6,070
12.	Chokepoint [173]	2011	64,204 images 54 videos	54	NR
13.	FG-NET [174]	2010	1,002	82	12.2
14.	YTC [175]	2008	1,910	47	NR
15.	LFW [41]	2007	13,000	5000	1/2.3/530

(*NR is Not Reported).

2.7 Summary of the Chapter

The current chapter describes conventional algorithms, deep learning-based approaches, transfer learning-based approaches, and ensemble learning-based approaches for face recognition. Limitations stemming from facial variations like lighting, pose variations, and expressions can affect the performance of conventional algorithms. Additionally, their ability to accurately recognize faces in uncontrolled settings is limited. The development of reliable automated face recognition systems utilizing computer vision techniques has revolutionized due to the rise of deep learning. With little pre-processing, multi-layer neural networks can detect visual features directly from image pixels. Deep learning's primary benefit is that it does not require manually created features. As part of the classifier learning process, it instead performs automated and optimized feature extraction, which does not compromise the correctness of face recognition.

The recent emergence of deep learning techniques has effectively addressed the limitations of conventional approaches. However, challenges persist in deep learning methodologies, including the reliance on extensive data and high-computing-power systems (e.g., GPUs for parallel processing). Acquiring substantial annotated facial datasets for facial recognition tasks remains problematic due to privacy concerns.

Ensemble learning can be employed by averaging the weights of multiple deep learning models to improve recognition accuracy. This approach amalgamates the advantages of deep learning and ensemble learning, culminating in enhanced generalization performance in the final model. The primary aim of this research is to create a streamlined framework that facilitates face recognition using minimal facial data and computational resources, all while maintaining a high level of accuracy. Consequently, the concept of deep ensemble transfer learning has been utilized to introduce an exceedingly efficient face recognition system rooted in deep learning principles. The detailed discussion of the proposed face recognition system is provided in the subsequent chapter of this thesis, followed by the chapter that introduces the data pre-processing techniques used in the presented research.

CHAPTER-3

DATASET PREPARATION AND PRE-PROCESSING

In this chapter, we delineated the strategies used to select the required datasets, including the freely available Internet sources we referenced for the self-curated dataset of mugshots. The creation of the dataset serves as the cornerstone for the entire research work and offers a comprehensive understanding of the methods used for data collection, acquisition, preprocessing, and curation. These processes were meticulously orchestrated to uphold the dataset's integrity, quality, and applicability within the study. This chapter also provides insight into the meticulous preprocessing steps known as data oversampling carried out to enhance the utilized datasets.

3.1 Datasets Used

We used four standard datasets (LFW, CPLFW, GT Face, and YTF) and one self-curated dataset to evaluate the efficiency of the HE-CNN model by considering different aspects of the evaluation.

LFW: Established in 2007, this dataset serves as a standard for face recognition and verification. It is publicly accessible and comprises 13,233 images depicting 5749 distinct identities. Among these, 1680 classes encompass multiple images, while 4096 classes feature only a single image. The images in the dataset are uniformly sized at 250x250 pixels, with a resolution of 96 DPI (Dots Per Inch), and are stored in Joint Photographic Experts Group (JPEG) format. The majority of the facial images exhibit a frontal orientation, specifically designed to address challenges like illumination shifts and partial occlusions [41].

CPLFW (Cross-Pose LFW): Released in 2018, this dataset is an upgraded iteration of the LFW standard dataset, specifically focusing on images showcasing substantial pose deviations. It encompasses a total of 11,652 images depicting 3928

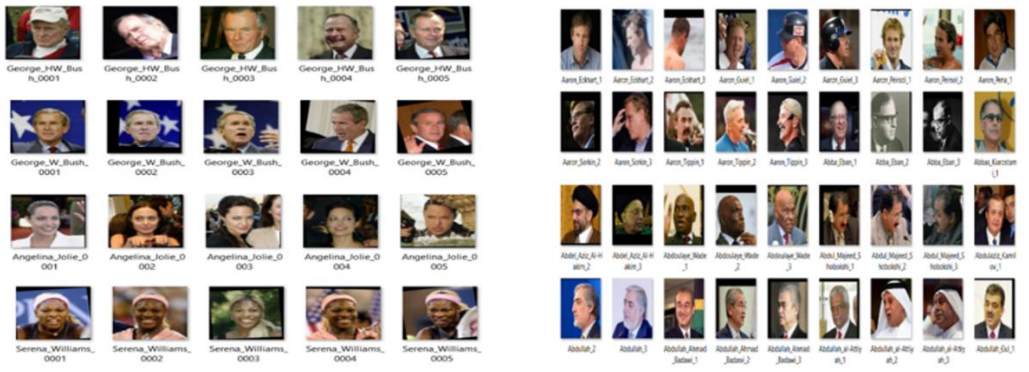
individuals, each class containing 2 or 3 images. Notably, the State-of-the-Art (SOTA) face recognition accuracy experiences a decline of 15-20% when applied to CPLFW in comparison to LFW. This discrepancy can be attributed to CPLFW's inclusion of images exhibiting pronounced variations in unconstrained factors [42]. The visual representations of LFW and CPLFW can be observed in Figure 3.1 (a) and Figure 3.1 (b), respectively.

GT Face: Photographs of 50 individuals were taken during two or three sessions spanning from June 1, 1999, to November 15, 1999, at the Center for Signal and Image Processing within the Georgia Institute of Technology. These images constitute the Georgia Tech face dataset. Each individual within the dataset is depicted through 15 color images captured in JPEG format. These images possess a resolution of 640x480 pixels and feature a busy background. On average, the facial regions in these images measure 150x150 pixels [43]. Notably, the faces contained in the dataset maintain a frontal orientation, as illustrated in Figure 3.2 (a).

YTF: It is a collection of facial videos that was curated to address the challenges associated with unconstrained facial recognition within videos. The dataset encompasses a total of 3,425 videos, featuring 1,595 distinct individuals. The source of these videos is YouTube. Each subject has about 2.15 videos, on average. The length of the video clip varies within the dataset, ranging from a minimum of 48 frames to a maximum of 6,070 frames. A typical video clip spans around 181.3 frames [44]. Sample frames from this dataset can be observed in Figure 3.2 (b).

Self-Curated Dataset: The primary objective of generating the self-curated dataset is to address the class imbalance inherent in LFW. The fact that the dataset's largest class contains 500 times more images than its smallest counterpart serves as an illustration of this imbalance. Such an imbalance can lead the model to exhibit bias towards the more abundant class. Additionally, the creation of this dataset serves to account for the presence of low-resolution images. Unlike standard datasets that

utilize high-resolution cameras, our dataset aims to incorporate real-world scenarios. To exemplify real-time face recognition applications, a self-curated dataset featuring 10 categories of criminals is formulated. Rigorous manual curation has been conducted to eliminate mislabeled or unclear images. Each category is represented by 25 meticulously chosen images to maintain a balanced distribution. From this pool, 10 images from each category are selected to craft images with reduced resolution and partially obscured faces. This results in 35 images per category within the created dataset. To ensure equitable representation, transformations are applied to individual class images, generating 50 augmented samples per class. Images for this self-generated dataset are sourced from Google (freely available images), specifically focusing on criminal subjects. Recognizing that video frames often capture multiple faces concurrently, an additional test dataset is constructed, encompassing 50 diverse images featuring multiple faces. This comprehensive dataset is now primed to demonstrate the viability of real-time surveillance systems. It is accessible for research purposes at the following link: https://data.mendeley.com/datasets/226275vfxz/2. Now, the meticulously devised dataset is structured to emulate real-world scenarios, incorporating variables such as partial occlusion, illumination shifts, pose variations, and low resolution, all of which are illustrated in Figure 3.3.



(a) (b)

Figure 3.1 Sample Images of (a) LFW and (b) CPLFW

Figure 3.2 Sample Images of (a) GT Face and (b) YTF Dataset

Figure 3.3 Sample Images of Self-Curated Dataset

3.2 Data Oversampling

Data augmentation, or data oversampling, serves as a strategy to amplify the dataset by generating virtual iterations of each image using a range of image transformation techniques. This practice is particularly valuable in the context of image classification tasks, as the augmentation process contributes to enhancing

model performance by providing a broader and more varied dataset [176]. In the presented research, we integrated data augmentation methodologies to rectify the inherent imbalances present in the datasets. This approach encompasses a diverse set of transformations, including but not limited to mirroring, rotation, shearing, cropping, zooming, and alterations to color saturation [114] [177].

Algorithm 3.1 specifies the complete process of oversampling the dataset in the present research. For every image, one augmentation technique (a_i) is randomly selected within configured thresholds, controlled by a parameter $'t'$, to apply a random magnitude of the transformation. If the dataset contains p samples in each class and the value of p varies within the classes of the dataset, then the proposed algorithm generates n target samples in each class to make it a class-balanced dataset. It selects the random augmentation a_i and applies all the r transformations to the randomly selected image. The output image I_{out} is generated after applying the transformations and added to the class C of the dataset. Figure 3.4 demonstrates a set of randomly generated oversampled images. Through the implementation of oversampling techniques, the datasets are equalized, leading to a uniform distribution of images across all classes. The quantity of images for oversampling is determined by selecting the maximum image count among all classes within the dataset. Some synthesized images of the self-curated dataset are also created by manually adding partial occlusion, illumination, and noise to the images, as shown in Figure 3.3. The standard class imbalanced datasets, such as the LFW and GT face datasets discussed in Section 3.1, have gone through oversampling techniques to make them balanced and increase the number of samples in each class.

Figure 3.4 Sample Output of Oversampled Images

Algorithm 3.1 Algorithm for the Process of Data Oversampling

Input: *Dataset D contains C different classes where each class consists of less than or equal to n unique samples, n is the maximum number of images in any class of the dataset, and A= [a1, a2, a3, a4, a5], where A is the set of all transformations applied on datasets.*
a1 = [centerCrop + ShiftScaleRotate + CLAHE]
a2 = [randomRotate90 + ShiftScaleRotate]
a3 = [flip + resize + randomBrightness]
a4 = [transpose]
a5 = [strongTransformation]
Output: *Dataset D contains C different classes where each class consists of n unique samples.*

1: **procedure** *Oversampling (D, n, A)*
2: n ⟵ target number of images per class
3: **for all** $C \epsilon D$ **do**
4: p ⟵ number of images in class C
5: **while** $p \leq n$ **do**
6: I ⟵ select one random image of C
7: a_i ⟵ select one random transformation function from A
8: r ⟵ transformations in a_i
9: I ⟵ I_{out}
10: **for all** $r \epsilon a_i$ **do**
11: I_{out} ⟵ $r(I_{out})$
12: **end for**
13: q ⟵ I_{out}
14: C ⟵ C U q
15: p ⟵ $p + 1$
16: **end while**
17: **end for**
18: **end procedure**

Here, a detailed discussion of the used data augmentation techniques and the application of these techniques in existing SOTA is given.

CenterCrop: The "CenterCrop" data augmentation approach isolates and extracts the central component of an image while excluding the surrounding areas. This method frequently improves the dataset's diversity and makes it easier to train machine learning models by concentrating on the image's most noticeable elements. The size of the original image is decreased, resulting in a new image that captures the core content by executing a center crop. This strategy is especially helpful when the principal object of interest is in the center and the surrounding context is less important. Center cropping can also help reduce unwanted noise or pointless details in the dataset, improving the generalization and performance of the model. Object recognition, classification, and segmentation are a few examples of image processing jobs that frequently use the center crop data augmentation technique [178] [179] [180]. Models are exposed to alterations in the main image content as a result of their application, which promotes resilience and adaptability in handling various scenarios and points of view.

ShiftScaleRotate: The technique of "ShiftScaleRotate" serves as a data augmentation method that introduces random affine transformations, including shifting, scaling, and rotating, to augment the training dataset. This approach diversifies the perspectives from which an object is observed within the dataset, enriching its variety. By incorporating these transformations, the dataset's diversity is heightened, bolstering the resilience and adaptability of machine learning models [181] [182]. Importantly, this augmentation strategy achieves these improvements without necessitating the acquisition and annotation of additional data points.

CLAHE: Contrast Limited Adaptive Histogram Equalization (CLAHE) data augmentation is a technique that employs adaptive histogram equalization in selected localized regions to enhance image contrast and detail. Contrary to

traditional histogram equalization, which may make the noise worse, CLAHE focuses on smaller image portions, preventing over-enhancing while maintaining the integrity of the entire image. This augmentation technique finds frequent application in enriching image datasets, especially within the realm of computer vision tasks. By introducing fluctuations in contrast and texture, it effectively heightens model resilience and performance [183] [184] [185].

RandomRotate90: The data augmentation method known as "RandomRotate90" introduces random rotations in 90-degree steps to images [186]. This method introduces variation and improves the model's performance and robustness by exposing the model to a variety of object orientations within the dataset. This augmentation method works exceptionally well for applications like object recognition, where there are a wide variety of object orientations. By including such rotations, the dataset becomes more inclusive and enables the model to generalize successfully across a variety of orientations without the need for additional labeled data.

Flip: The "Flip" technique is a data augmentation approach that entails mirroring images horizontally or vertically. This strategy enriches dataset diversity by showcasing objects in altered orientations or viewpoints. Horizontal flipping entails a left-to-right reversal, whereas vertical flipping involves an up-to-down reversal. Frequently employed in image processing and computer vision applications, flipping serves to enhance model generalization and overall performance, particularly in tasks where object orientation is of secondary importance [187] [188]. Through the integration of these mirror-image alterations, the dataset gains breadth, thereby enhancing the model's proficiency in detecting and comprehending objects across diverse vantage points.

Resize: The technique of "Resizing" in data augmentation involves adjusting the dimensions of images while preserving their original aspect ratio. This method

modifies image size within a dataset, either increasing or decreasing resolution and it is widely employed. The versatility of resizing enables images to be standardized to meet specific input size prerequisites for machine learning models or to introduce size variations for enhanced generalization. During resizing, images can undergo enlargement or reduction, impacting the level of detail and potentially accentuating distinct attributes. This method proves especially valuable when handling images of disparate dimensions within a dataset or when preparing data to align with precise model input specifications. Employing resizing as a data augmentation approach renders the dataset adaptable to the model's requirements, fostering heightened performance and accuracy during both training and testing stages [189] [190].

RandomBrightness: The data augmentation approach known as "Random Brightness" encompasses the application of random changes to the brightness levels within images. Through this method, fluctuations in illumination are introduced, bolstering the dataset's resilience and capacity to capture distinct lighting situations. By incorporating random brightness adjustments, the augmentation procedure emulates real-life settings characterized by shifting lighting conditions. Consequently, the model's capacity to generalize effectively and achieve robust performance across a spectrum of environments is heightened. Notably beneficial for tasks like object detection and recognition, where objects can manifest amidst diverse lighting scenarios, this technique proves to be an invaluable asset [191] [192].

Transpose: It serves as a data augmentation technique involving the exchange of rows and columns within an image matrix. The introduction of transpose augmentation injects diverse spatial layouts of objects and patterns, enriching the dataset's variety. Through the application of transpose, the dataset gains an assortment of alternate perspectives for the same image, permitting the model to glean insights from varying spatial correlations. This augmentation method is

especially advantageous in endeavors like image classification and pattern recognition, wherein object orientations may differ. The process of transposition gives the model the ability to handle changes in how objects are aligned and how they are arranged in space. This allows the model to be more general and perform better [193].

StrongTransformation: In the context of data augmentation, a "Strong Transformation" refers to a more profound and impactful alteration applied to an image. This category of transformation frequently encompasses a fusion of various augmentation techniques, including rotations, flips, adjustments in brightness, contrast, and other modifications. The utilization of strong transformations aims to introduce substantial diversity into the dataset, compelling the model to navigate through a spectrum of scenarios and amplifying its capacity to withstand shifts in real-world conditions. Such transformations prove especially advantageous during the training of models intended to navigate intricate and heterogeneous environments, exposing the model to an extensive array of potential inputs and circumstances [194].

These techniques are combined and used in different ways in our work to create a large dataset that can improve the performance of the model.

3.3 Summary of the Chapter

In conclusion, the dataset preparation chapter is a thorough walkthrough of the complex process of converting unstructured, low-quality raw data into a well-organized, high-quality dataset. It demonstrates the commitment to transparency and rigor in determining the dataset's suitability for answering the research questions and serves as a crucial preface to the next analytical chapters, establishing the foundation for insightful deductions and significant discoveries. Data oversampling techniques have been applied to address the challenges posed by imbalanced classes within datasets and instances where certain classes are

underrepresented due to a limited number of images. The self-made dataset has been created to demonstrate the real-time application of the present research. The present work used four standard datasets and one self-curated dataset for the evaluation of the implemented system that is going to be discussed in the next chapter of the thesis.

CHAPTER -4

A NOVEL METHOD FOR AUTOMATIC FACE RECOGNITION SYSTEM

In this chapter, a detailed overview of the proposed automated face recognition system and HE-CNN model has been discussed. The proposed and implemented modifications to the baseline models are also discussed in sub-sections of this chapter. The face detection in an automated system is done using SSD as it is faster and accurate in comparison to other existing face detection algorithms.

4.1 The Proposed Automated Face Recognition System

This section proposes an automated system for criminal face identification and also helps police officials identify crime-prone areas. Figure 4.1 depicts the graphical representation of the proposed automated recognition system that comprises face capture, face detection, face recognition, alert generation, and prediction of crime-prone regions. Algorithms 4.1, 4.2, 4.3, and 4.4 demonstrate the workings of an automated recognition system proposed in the present research. The methodology in the present research is divided into four modules: database creation, criminal recognition, alert generation, and prediction of crime-prone areas.

In the provided algorithms, the video is captured from the Global Positioning System (GPS)-enabled camera that attaches the location coordinates $L = \{lat, long\}$ with the video frames $D = \{F_i, L\}_{i=1}^{n}$, where F_i is the i^{th} frame, and L is the location coordinates of the camera. The GPS is used in the presented solution to track the current location of the static cameras that are deployed in different locations of the city. Then, the number of detected and aligned faces $\{\{A_1, L\ \}, \{A_2, L\ \}, \ldots., \{A_y, L\}\}$ from the captured video frames using SSD is stored in A along with the location coordinate L. The detected and aligned faces with the

location coordinates are transferred to the recognition module for the recognition of criminals using the HE-CNN model. The recognized faces $\{\{R_1, L\}, \{R_2, L\}, \ldots, \{R_y, L\}$ and the location coordinates are sent to the alert generation phase. The system finds out the distance between the police stations $P = \{\{lat_1, long_1\}, \{lat_2, long_2\}, \ldots, \{lat_z, long_z\}\}$ and the location of the criminal $L = \{lat, long\}$ using the Haversine formula [195]. After calculating the distance, $Ð = min(H_j)\ _{j=1}^{z}$ stored the distance from the nearest police station. Then, the implemented system generated an alert via message $\{I_i, L\}_{i=1}^{y}$ and e-mail $\{R_i, I_i, L\}_{i=1}^{y}$ to the registered contact number and e-mail ID of the nearest police station, where I_i is the information (name, age, gender, crime, date of crime, identity mark) of the i^{th} criminal, L is the location of the criminal, and R_i is the image of the criminal. Parallelly, the location coordinates of the identified criminal L= $\{lat, long\}$ is stored in a separate file to form clusters of crime-prone regions. $L' = \{\{lat', long'\}, \{lat'', long''\}, \ldots, \{lat^{y'}, long^{y'}\}\}$ contains the location coordinates of identified criminals collected through the cameras installed in different locations. The clusters $C = \{c_i\}_{i=1}^{p}$ are formed using the K-means clustering technique [196]. These clusters are then visualized on a Google Map, as the traffic flow in different areas is represented. This information serves as a valuable tool for police officials to identify regions that are prone to criminal activity. The input and output of the implemented system are given below. The results produced by Algorithm 4.1 serve as the input for Algorithm 4.2. Likewise, Algorithm 4.2's output is utilized as input for Algorithm 4.3. Algorithm 4.4, on the other hand, receives input in the form of location coordinates for all identified criminals, which are then used to create clusters in crime-prone regions.

Input: Video frames from the GPS-enabled camera $D = \{F_i, L\}_{i=1}^{n}$ and location coordinates of registered police stations $P = \{lat_i, long_i\}_{i=1}^{z}$

//where F_i is the i^{th} frame, n is the total number of frames, $L = \{lat, long\}$ is the location coordinates (*i.e.*, latitude and longitude) of the camera that is the same for all the frames, and z is the number of registered police stations.

Output: Message and e-mail containing image and information about the criminal $M = \{m_{i,}\ e_i\}_{i=1}^{y}$ and clusters of the crime-prone regions C=$\{c_i\}_{i=1}^{p}$

//where y is the number of criminals identified, and p is the number of clusters formed.

Algorithm 4. 1 Face Detection

Input: *Video frames from the GPS-enabled camera* $D = \{F_i, L\}_{i=1}^{n}$ *and location coordinates of registered police stations* $P = \{lat_i, long_i\}_{i=1}^{z}$
//where F_i *is the* i^{th} *frame, n is the total number of frames,* $L = \{lat, long\}$ *is the location coordinates (i.e., latitude and longitude) of the camera that is the same for all the frames, and* z *is the number of registered police stations*
Output: *Detected faces with location coordinates {Aᵢ, L} of* y *criminals*

1. **procedure** *FaceDetection (D)*
2. $D = \{\{F_1, L\}, \{F_2, L\}, \ldots., \{F_n, L\}\}$
3. **Repeat**
4. **for all** $\{F_i, L\} \epsilon$ **D do**
5. SSD ⟵ $\{F_i\}$
6. A ⟵ $\{\{A_1, L\}, \{A_2, L\}, \ldots., \{A_y, L\}\}$
7. **end for**
8. **until** n **times**
9. **return** $\{A_i,\ L\}_{i=1}^{y}$
10. **end procedure**

Algorithm 4.2 Face Recognition

Input: *Detected faces and location coordinates of criminals from algorithm 4.1(i.e., {Aᵢ, L})*
Output: *Recognized faces with location coordinates {Rᵢ, L} of* y *criminals*

1. **procedure** *FaceRecognition (A)*
2. $A = \{\{A_1, L\}, \{A_2, L\}, \ldots., \{A_y, L\}\}$
3. **Repeat**
4. **for all** $\{A_i, L\} \epsilon A$ **do**
5. HE-CNN ⟵ $\{A_i\}$
6. R ⟵ $\{\{R_1, L\}, \{R_2, L\}, \ldots., \{R_y, L\}$
7. **end for**

8. **until y times**
9. **return** $\{R_i, L\}_{i=1}^{y}$
10. **end procedure**

Algorithm 4.3 Alert Generation

Input: *Recognized faces and location coordinates of criminals from algorithm 4.2 (i.e., {R_i, L}) and registered police stations records P*
Output: *Message and e-mail containing image and information about the criminal (i.e.,* $M = \{m_{i,} e_i\}_{i=1}^{y}$

1. **procedure** *AlertGeneration (R, P)*
2. *// P is the police station record that contains the contact details, e-mail ID, and latitude and longitude of all registered police stations*
   $R = \{\{R_1, L\}, \{R_2, L\}, \ldots, \{R_y, L\}\}$
3. $L = \{lat, long\}$
4. $P = \{\{lat_1, long_1\}, \{lat_2, long_2\}, \ldots, \{lat_z, long_z\}\}$
5. **Repeat**
6. **for all** $\{R_i, L\} \epsilon R$ **do**
7. **Repeat**
8. **for all** $\{P_j\} \epsilon P$ ***do***
9. $H_j \longleftarrow \{L, P_j\}_{j=1}^{z}$
10. $Ð \longleftarrow Min(H_j)_{j=1}^{z}$
11. **end for**
12. **until z times**
13. $m_i \longleftarrow \{I_i, L\}_{i=1}^{y}$
14. $e_i \longleftarrow \{R_i, I_i, L\}_{i=1}^{y}$
15. **end for**
16. **until y times**
17. **return** $\{m_{i,} e_i\}_{i=1}^{y}$
18. **end procedure**

Algorithm 4.4 Clusters of Crime Prone Regions

Input: *Location coordinates of the recognized criminals*
Output: *Clusters of the crime-prone regions C=*$\{c_i\}_{i=1}^{p}$

1. **procedure** *Cluster (L)*
2. // L' stores the location coordinates (L) of all the identified criminals from the cameras installed in different locations
   $L' = []$
3. **Repeat**
4. **for** $k \longleftarrow$ ***1 to y*** **do**

5. $(lat^{k\prime}, long^{k\prime})= L$
6. $L'= L'+ (lat^{k\prime}, long^{k\prime})$
7. **end for**
8. **until y times**
9. // Therefore, $L'= \{\{lat', long'\}, \{lat'', long''\}, ..., \{lat^{y\prime}, long^{y\prime}\}\}$
   $\{c_1, c_2, \dots, c_p\} \longleftarrow$ *K-means(L′)*
10. $C=\{c_i\}_{i=1}^{p}$
11. **return *C***
12. **end procedure**

4.2 The Proposed Modified Architecture of Baseline Models

In this section, various pre-trained models such as ResNet50, VGG19, and DenseNet169 have been used for the present work. These models were fine-tuned and combined through ensemble transfer learning to create an optimized hybrid model specifically designed for the task at hand. The proposed modification consists of a baseline model and customized classification layer. For the base network, the pre-trained ResNet50, VGG19, and DenseNet169 models, using their initial weight parameters have been utilized. The base architecture of pre-trained models is originally trained on the ImageNet dataset and includes 1000 columns of distinct weight matrices at the end [64]. However, these weight matrices are not significant for our experiments, as the classes in the face datasets differ from those in the ImageNet dataset. To adapt the model to our task, we introduced two new weight matrices in the classification head section with a Leaky ReLU activation function. We employed Kaiming initialization to initialize these weight matrices [197]. Kaiming initialization has been utilized to prevent the activation outputs of the layers from exploding during the forward pass in a deep neural network. At each layer *'l'*, the weight matrix is initialized with random numbers drawn from a standard normal distribution, where each random number is multiplied by the value of *'fan_in'*, representing the number of input connections or the number of neurons in the previous layer that connect to the current layer. It indicates the size of the input space for a specific layer *'l'*. Further, *'fan_out'* refers to the number of output

connections, or the number of neurons in the current layer that connect to the next layer. It represents the size of the output space for a specific layer. Therefore, in the present work, we replaced Xavier initialization [198], where the weights of a layer are initialized using random values selected from a uniform distribution with specific bounds, as shown in equation (4.1).

$$SD = \frac{\sqrt{2}}{\sqrt{fan\_in+fan\_out}} \tag{4.1}$$

Here, SD is the Standard Deviation of the random numbers drawn from a standard normal distribution, which are used to initialize the weights of a layer. The initial layers of the CNN model are responsible for extracting features, while the final layers are utilized for classification purposes. The compact representations of the pre-trained models used in this study (VGG, ResNet, and DenseNet) are illustrated in Figure 4.2. However, based on experimental findings in the field of facial recognition algorithms, relying solely on pre-trained models is insufficient to achieve optimal accuracy. Therefore, certain modifications have been implemented to enhance the recognition accuracy of these models. In this work, a modified architecture for the base models is introduced, which involves the incorporation of global pooling, batch normalization (BN), and dropout in the classification layers. The addition of a pooling layer helps reduce the number of trainable parameters in the model. Typically, two types of pooling techniques have been employed, namely average pooling and max pooling, which can be mathematically described using equations (4.2) and (4.3).

$$P = O_{max}^{n,n}(F) \tag{4.2}$$

$$P = O_{avg}^{n,n}(F) \tag{4.3}$$

In the present work, the input feature map F obtained from the previous convolutional layer has been processed using pooling operations. The maximum pooling operation, denoted as $O_{max}^{n,n}(F)$ operates on the input feature map of

size $n \times n$, while the average pooling operation, denoted as $O_{avg}^{n,n}(F)$, calculates the average value. The output of the pooling layer, denoted as *P*, is obtained by concatenating the maximum and average values using the concatenate function in the Keras library. Both the maximum and average pooling techniques have their advantages, and their performance can vary depending on the activation map's maximum and average values. To preserve both of these values, the concatenation technique has been employed. Global pooling is used to reduce each channel in a feature map to a single value and serves as an alternative to densely connected or fully connected layers in a classifier. It helps reduce the model's complexity. Batch normalization has been utilized to normalize the positive and negative features from the previous convolutional layer, addressing the issue of covariate shift [122]. It effectively improves accuracy without any side effects [199]. To prevent overfitting, dropout layers have been added for regularization. Dropout is a regularization technique that randomly drops out a fraction of the neurons during training. The optimal dropout value for the model is determined experimentally, as it can significantly impact the model's accuracy [199]. The non-linearity functions, such as ReLU [200], PReLU [197], Leaky ReLU [201], *etc.*, can be placed before or after the BN layer. In the present work, the use of Leaky ReLU after the BN layer gives better results. Therefore, we used the Leaky ReLU activation function because it alleviates the problem of "dying ReLU" [201]. The mathematical expression for calculating the value of Leaky ReLU is provided in equation (4.4).

$$f(x) = \max(0.01 * x, x) \tag{4.4}$$

The Leaky ReLU activation function is defined as follows: when given a positive input *x*, it produces a value of *x*; however, if the input is negative, it outputs a minimum value of 0.01 times *x*. This modification allows Leaky ReLU to produce an output for negative inputs as well. Unlike the standard ReLU function, this modification results in a non-zero gradient on the left side of the mathematical graph, effectively addressing the issue of "dead neurons" in that region. The

modified architecture of the baseline models, incorporating Leaky ReLU and other enhancements, is depicted in Figure 4.3. The reasons for these modifications in the baseline model architecture are discussed in Table 4.1, outlining the benefits and justifications for each modification.

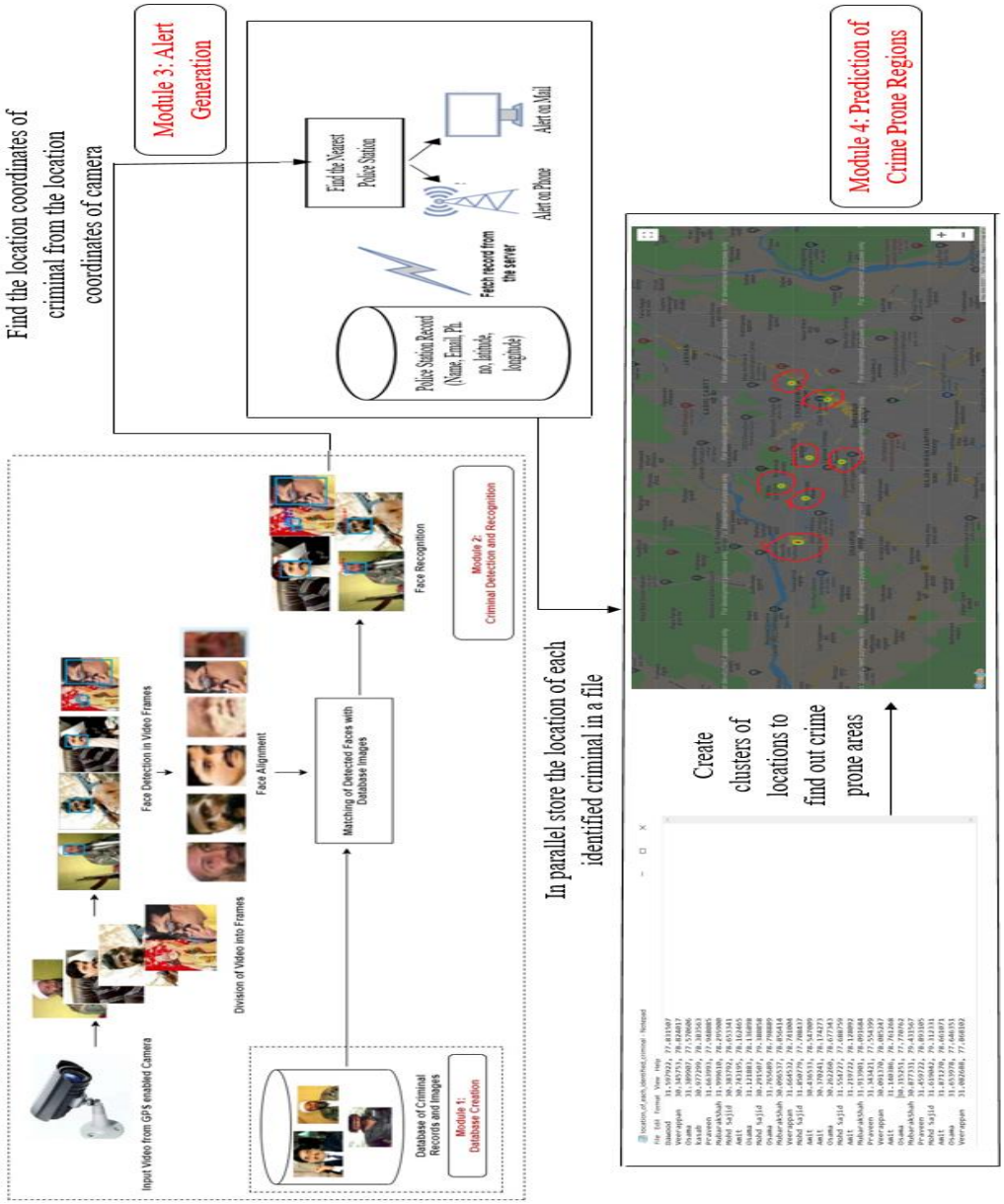


Figure 4.1 The Schematic Flow of the Proposed Automated Face Recognition

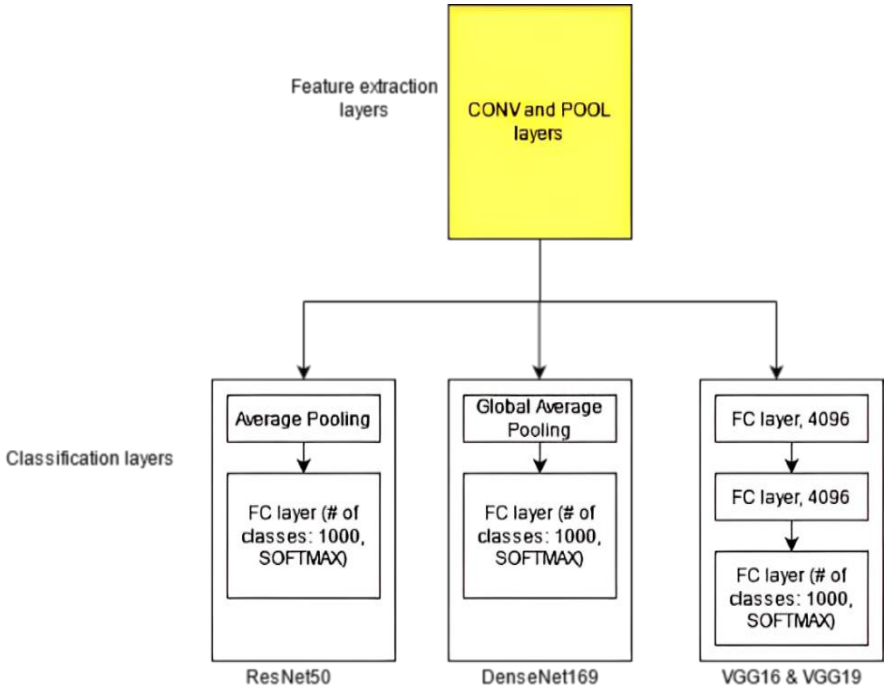


Figure 4.2 The Architecture of Classification Layers of Pre-Trained Models (ResNet50, DenseNet169, VGG16, and VGG19)

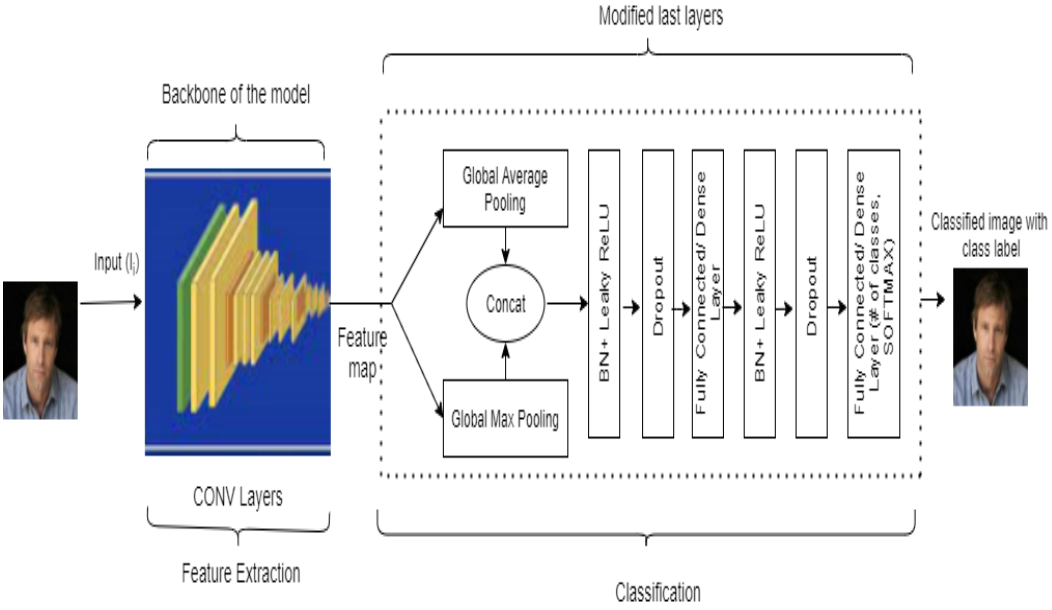


Figure 4.3 The Modified Architecture of the Baseline Model Consisting of GMP, GAP, BN, dropout, and FC layers (The Dotted Line Shows the Modified Part of the Model)

Table 4.1 The Persuasive Reasons for the Rectification of the Classification Layers of Baseline Models

S. No.	Charact-eristics	Standard ResNet50	Standard VGG19	Standard DenseNet 169	Modified Architecture	Persuasive Reasons for the Modifications
1.	Pooling layer	Average pooling	No pooling layer	GAP	Concatenation of GAP and GMP	The activation map from the previous layer can outperform its mean value, and vice versa.
2.	No. of FC layer	1	3	1	2	Adding a layer enhances ResNet and DenseNet, while removing one improves VGG.
3.	Linear activation	ReLU	ReLU	ReLU	Leaky ReLU	To address the issue of dying ReLU.
4.	Regulariz-ation	No dropout	No dropout	No dropout	The dropout layer is used	To minimize the overfitting issue.

4.2.1 The Proposed Modified DenseNet169 Model for Face Recognition

Dense Convolutional Neural Network proposed by Huang *et al.* [72] comprises a convolution and pooling layer, transition layers, dense blocks, and a classification layer. The convolution layer holds the filters to be applied to the feature map, whereas the pooling layer helps minimize the dimension of the feature map. The dense block is utilized to connect all the layers in such a way that each layer receives input from all the preceding layers. The current layer concatenates the features and passes its own feature maps to all the subsequent layers. The addition of dense blocks increases the number of channels, resulting in a complex model. Therefore, the transition layer helps to control the complexity of DenseNet. The concept of skip connections, similar to residual networks [70], is also utilized to enhance the performance of the network without increasing its depth. The problem of vanishing gradient [202] is successfully circumvented in DenseNet by the use of skip connections. In CNN, two layers l and $l-1$ are connected using the composite

function *F*, which consists of the convolution layer, pooling layer, batch normalization, and Rectified Linear Unit (ReLU). The outcome of the preceding layer X_{l-1} is considered input to the next layer X_l, as represented in equation (4.5).

$$X_l = F(X_{l-1}) \tag{4.5}$$

However, the layers 0, 1, 2, …. , $l-1$ are connected in the dense block in such a way that the concatenation of the outputs $[X_0, X_1, X_2, \dots, X_{l-1}]$ of all the layers is passed as input to the subsequent layer, as represented in equation (4.6). In this way, a layer acquires the aggregate information of all the preceding layers. Hence, dense convolutional networks are named due to the dense connectivity between the layers in the network.

$$X_l = F([X_0, X_1, X_2, \dots, X_{l-1}]) \tag{4.6}$$

The present work proposes the improved architecture of DenseNet169 by adding global pooling, batch normalization (BN), and dropout in the classification layers of the model. The modified architecture can be mathematically expressed using equations (4.7) – (4.15), and a diagrammatic representation is given in Figure 4.4.

$$X_6 = F([X_0, X_1, X_2, X_3, X_4, X_5]) \tag{4.7}$$

$$X_6 = \text{TL}(X_6) \tag{4.8}$$

$$X_{18} = F([X_6, X_7, \dots, X_{17}]) \tag{4.9}$$

$$X_{18} = \text{TL}(X_{18}) \tag{4.10}$$

$$X_{50} = F([X_{18}, X_{19}, \dots, X_{49}]) \tag{4.11}$$

$$X_{50} = \text{TL}(X_{50}) \tag{4.12}$$

$$X_{82} = F([X_{50}, X_{51}, \dots, X_{81}]) \tag{4.13}$$

$$O = \textit{Concat (GAP}(X_{82})\textit{, GMP}(X_{82})\textit{)} \tag{4.14}$$

$$C = H(O) \tag{4.15}$$

Here, $[X_0, X_1, \dots, X_{82}]$ is the concatenation of the outputs of the layers in the dense block. TL is the transition layer applied to the output received from the dense block consisting of BN, ReLU, and 1x1 Convolution layer followed by global average pooling. F is a composite function comprising BN, ReLU, 1x1 Convolution layer followed by BN, ReLU, and 1x1 Convolution layer. Moreover, the *Concat* operation denotes the concatenation of GAP and GMP. H is a function containing BN, Dropout, Leaky ReLU, and two fully or densely connected layers succeeded by logarithmic SOFTMAX. The output of H is the number of classes represented as C.

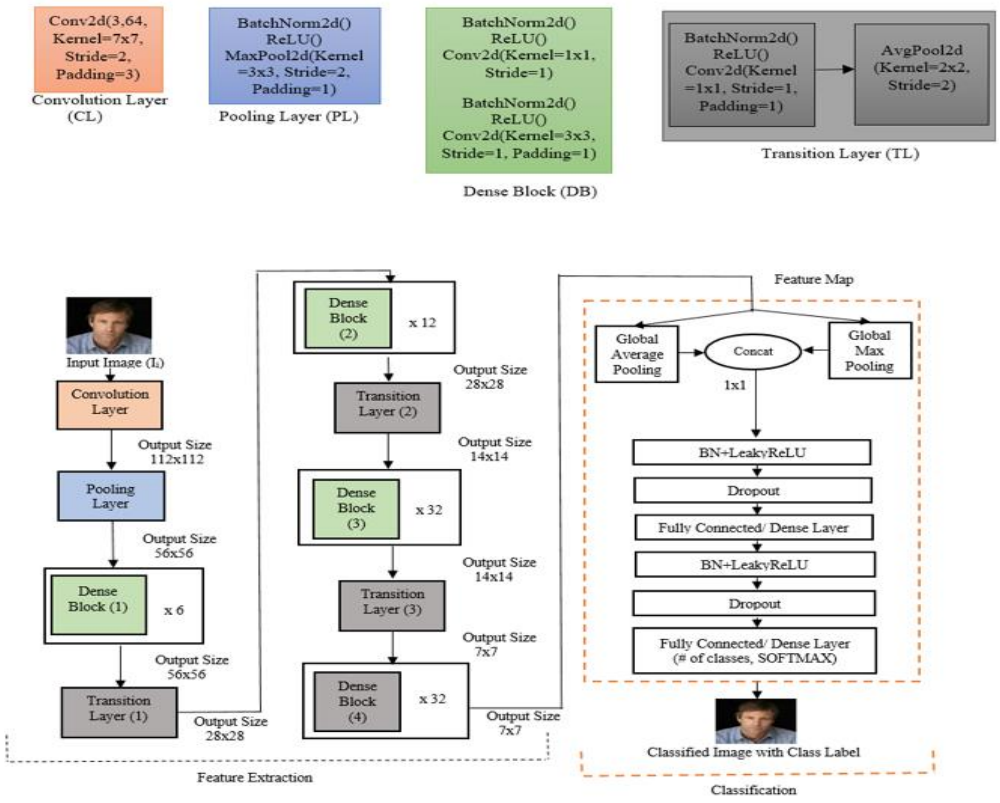


Figure 4.4 The Proposed Modified Architecture of DenseNet169 Consisting of GMP, GAP, BN, dropout, and Dense/ Fully Connected Layers (Orange Dotted Line Shows the Modified Part of the Model).

4.2.2 The Proposed Modified VGG19 Model for Face Recognition

The deep Convolutional Neural Network (VGGNet), specifically the VGG19 configuration, is employed in this study. VGG19 encompasses 19 layers: 16 convolutional layers and 5 pooling layers for the feature extraction phase, followed by 3 fully connected layers for classification purposes [67]. Within the convolutional layer, diverse filters are applied to the feature map, while the pooling layer maintains the map's dimensions. During training, the convolutional layers process RGB images of a consistent size, with dimensions of 224 × 224. Pre-processing involves subtracting the calculated mean RGB value from each pixel in the image. The image then traverses through a series of convolutional (CONV) layers employing 3x3 filters, a fixed stride of 1, and spatial padding. Max-pooling is executed over a 2x2-pixel window with a stride of 2. Subsequent to the convolutional layers, three Fully-Connected (FC) layers come into play. The first two FC layers comprise 4096 channels each, while the final FC layer consists of 1000 channels, aimed at classifying each class in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) dataset. The last layer is the Softmax layer in the architecture.

A composite function *F*, encompassing convolutional layers, pooling layers, Rectified Linear Units (ReLU), and Batch Normalization (BN), serves to link two adjacent layers, denoted as l and $l-1$. Notably, X_l is reliant on the output of X_{l-1}, as illustrated in equation (4.16). In the context of VGG19, the range for l spans from 1 to 5, corresponding to the five blocks $[X_1, X_2, \dots, X_5]$ that delineate the complete architectural structure.

$$X_l = F(X_{l-1}) \tag{4.16}$$

The outcome of each block serves as input for the subsequent block. The ultimate output feeds into the stack of FC layers, denoted by function *D*, comprising three fully connected layers. Following these layers is the application of logarithmic

Softmax, as depicted in equation (4.17). The result, represented as *C*, stems from the output of function *D* and signifies the count of classes.

$$C = D(X_5) \tag{4.17}$$

The modifications to the architecture involve the incorporation of global pooling, batch normalization, and dropout into the pre-existing classification layers, aimed at refining the VGG19 architecture. The adapted VGG19 architecture is visually outlined in Figure 4.5, while its mathematical elucidation is detailed in equations (4.18) – (4.20).

$$X_5 = F(X_4) \tag{4.18}$$

$$\hat{O} = \mathcal{F}(X_5) \tag{4.19}$$

$$¢ = \mathrm{H}(\hat{O}) \tag{4.20}$$

The output of block 4 (denoted as X_5) undergoes processing through the function $\mathcal{F}$, wherein $\mathcal{F}$ is formed by concatenating Global Average Pooling (GAP) and Global Max Pooling (GMP). This amalgamation is mathematically depicted in equation (4.21).

$$\mathcal{F}(X_5) = \mathrm{GAP}(X_5) \oplus \mathrm{GMP}(X_5) \tag{4.21}$$

In this context, the symbol $\oplus$ signifies the fusion of global max pooling (gmp) and global average pooling (gap), while H stands for a composite function that encompasses Batch Normalization (BN), Dropout, and Leaky ReLU, along with two fully connected layers, culminating in logarithmic Softmax. The result, ¢ emanates from the output of H and corresponds to the count of classes.

4.2.3 The Proposed Modified ResNet50 Model for Face Recognition

A deep neural network architecture, ResNet50, proposed by He *et al.* [70] consists of 50 layers and comprises 4 stages, represented as S_1, S_2, S_3 and S_4. It uses residual blocks and identity blocks to overcome the vanishing gradient problem,

which is a common issue in deep neural networks that can prevent effective training. To begin the process, the input image undergoes an initial convolution (I_c) using a kernel size of 7x7. Subsequently, a max-pooling layer with a 3x3 kernel size and a 2-stride is applied. This leads to a reduction in the image's width and height to one-fourth of its original dimensions, while the channel size is augmented to 64. Each of the four stages comprises two residual blocks and one identity block. The residual block serves as a foundational unit, introducing skip connections to facilitate smoother gradient flow throughout the network. It encompasses two convolutional layers, followed by a skip connection that adds the input to the output of the second convolutional layer. This skip connection enables the gradient to circumvent the convolutional layers, directly progressing to the subsequent layer, thereby preventing the gradient from vanishing. In comparison, the identity block represents a specialized version of the residual block, where the input and output share identical dimensions. It encompasses three convolutional layers: the first and third layers possess a 1x1 kernel size and a stride of 1, while the second layer employs a 3x3 kernel size and a stride of 1 or 2, contingent on the input size. The skip connection in the identity block is a simple identity mapping, which allows the gradient to flow more easily through the network.

Average pooling and one Fully-Connected (FC) layer is used after a stack of convolutional layers. The FC layer consists of 1000 channels to classify each class of the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) dataset. The last layer is the Softmax layer in the architecture. A composite function F containing the convolution layers as well as the pooling layer, along with the Rectified Linear Unit (ReLU) and Batch Normalization (BN), is used to connect two layers, l and $l - 1$. Here, S_l uses the output from S_{l-1} as shown in equation (4.22). In ResNet50, l is defined from 1 to 4, as we have four such stages $[S_1, S_2, \dots, S_4]$ to represent the entire architecture.

$$S_l = F(S_{l-1}) \tag{4.22}$$

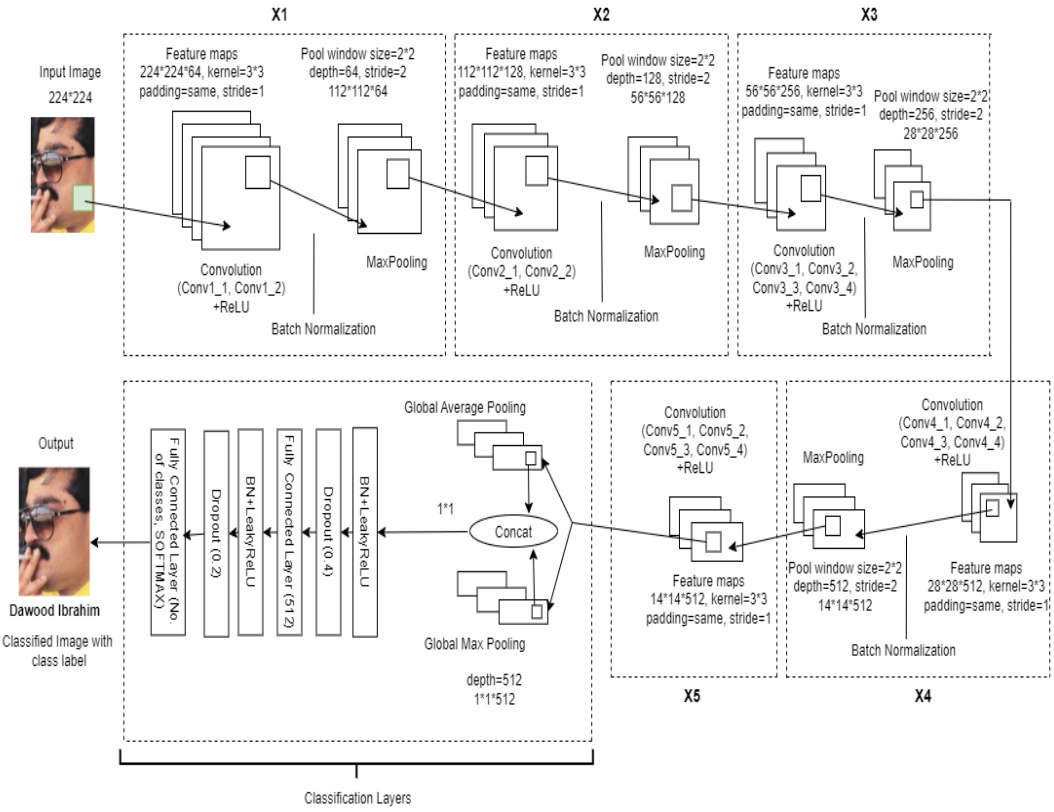


Figure 4.5 The Architecture of the Proposed Modified VGG19

The output of every block is passed as input to the next block. The last output of S_4 is passed to the stack of average pooling and FC layer represented by a function D followed by logarithmic Softmax as shown in equation (4.23). C is the output of the function that represents the number of classes.

$$C = (S_4) \tag{4.23}$$

The modification in the architecture is done by the addition of global pooling, batch normalization, and dropout in the existing classification layers in order to improve the architecture of ResNet50. The modified architecture of ResNet50 has been represented in Figure 4.6, and the mathematical explanation is expressed through equations (4.24) – (4.26).

$$S_4 = F(S_3) \tag{4.24}$$

$$€ = ʃ\,(S_4) \tag{4.25}$$

$$¢ = \; ʜ(€) \tag{4.26}$$

The output of stage 4 ($i.e., S_4$) is processed through the ʃ function, where ʃ consists of the concatenation of Global Average Pooling (GAP) and Global Max Pooling (GMP), mathematically represented using equation (4.27).

$$ʃ(S_4) = \text{GAP}(S_4) \oplus \text{GMP}(S_4) \tag{4.27}$$

Here, the symbol ⊕ denotes the concatenation of gmp and gap, and ʜ represents a function that consists of BN, Dropout and, Leaky ReLU, along with two fully connected layers that are followed by logarithmic Softmax; ¢ is the output of ʜ *i.e.*, the number of classes.

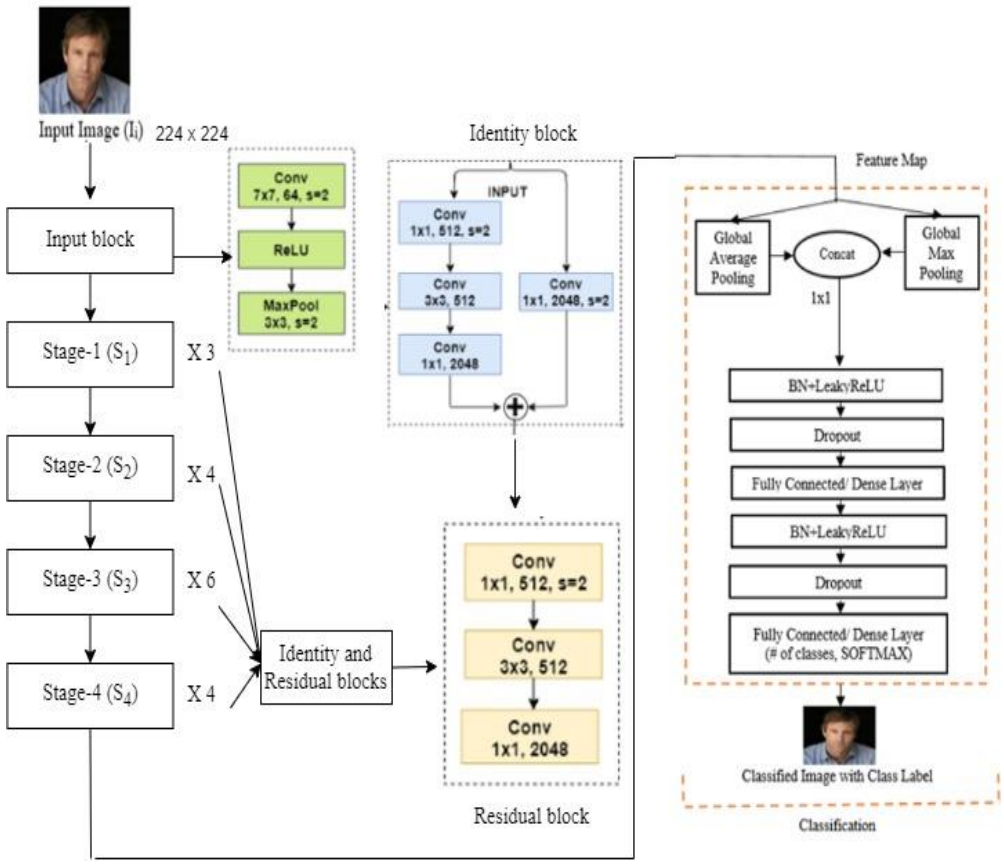


Figure 4.6 The Architecture of the Proposed Modified ResNet50

4.3 The Proposed and Implemented Novel Hybrid Ensemble CNN (HE-CNN)

The use of ResNet, VGG, and DenseNet in an ensemble model for face recognition can lead to improved accuracy, robustness, diversity, and flexibility in the model design. ResNet, VGG, and DenseNet are all deep neural network models that have been widely used for image classification tasks [203] [204] [205]. Each of these models boasts a distinct architecture designed to encapsulate varied features and patterns inherent in an image. Through amalgamating their outputs, the ensemble model attains superior accuracy in contrast to any standalone model. Ensemble models exhibit heightened resilience against overfitting and a greater propensity to generalize well on novel data. By melding disparate models, the ensemble can glean insights from a broader spectrum of features and patterns, cultivating robustness against input variations. ResNet, VGG, and DenseNet each present distinct architectural nuances. Consequently, integrating them within an ensemble introduces diversity to the models, thereby fostering enhanced overall performance. Specific strengths and weaknesses in feature extraction are intrinsic to each model. For instance, ResNet might excel in capturing edge and corner features, whereas DenseNet might prove adept at discerning intricate patterns within textures. We can take advantage of their complementary strengths by combining the models. Using an ensemble allows for more flexibility in the design of the model. By adjusting the weights assigned to each model, we can optimize the ensemble to achieve the desired level of accuracy, efficiency, and resource utilization. Therefore, the ensemble of the modified versions of these three models has been used to get an optimized hybrid model for the face recognition task.

Algorithm 4.5 adapts the optimized baseline models and employs an ensemble learning strategy to craft an effective hybrid model. The stacking method, a hallmark of ensemble learning, has employed to devise this hybrid model tailored for the face recognition task. The ultimate prediction of the hybrid model is realized

by aggregating outcomes from the fine-tuned baseline models through a weighted sum operation. This operation, commonly known as weighted average [206], is often employed in ensemble models to accord greater significance to predictions from better-performing models. The idea behind the weighted sum operation is to give different weights to the predictions of each model in the ensemble based on how well they did on a validation set. Models exhibiting superior performance receive higher weights, while those performing less effectively receive lower weights. As a result, the ensemble's final prediction bears a more pronounced influence from the better-performing models and a diminished influence from the weaker ones. Applying the weighted sum operation to ensemble models allows us to harness the strengths of multiple models while mitigating their shortcomings. This approach fosters superior overall performance and more resilient predictions [207]. The predicted face using the hybrid ensemble model, denoted as PHE−CNN, is defined by equation (4.28). Through experimentation, VGG19 showcases superior accuracy. Consequently, the VGG19 version is chosen over the VGG16 counterpart. Nonetheless, the present work incorporated VGG19, DenseNet169, and ResNet50 within the hybrid model. The comprehensive procedure is visually outlined in Figure 4.7.

$$P_{HE-CNN} = \sum_{i=1}^{\S} W_i \sum_{j=1}^{n} \frac{1}{\sum_{k=1}^{C} e^{\theta_k^T f^{(j)}}} \begin{pmatrix} e^{\theta_1^T f^{(j)}} \\ \dots \\ e^{\theta_C^T f^{(j)}} \end{pmatrix} \tag{4.28}$$

Here W_i denotes the weight of modified baseline models, § is 3 because the present work considered three models for the ensemble model, n is the count of image samples in training data, C denotes the number of classes in a dataset, $f^{(j)}$ is the feature of the j^{th} sample, θ is the parameter matrix of the softmax loss function $L(\theta)$, and $\theta_k^T f^{(j)}$ denotes the inner product of θ_k and $f^{(j)}$. The optimal values of W_1, W_2, and W_3 are selected using the VotingClassifier available in the scikit-learn library of Python (https://rb.gy/p0ig).

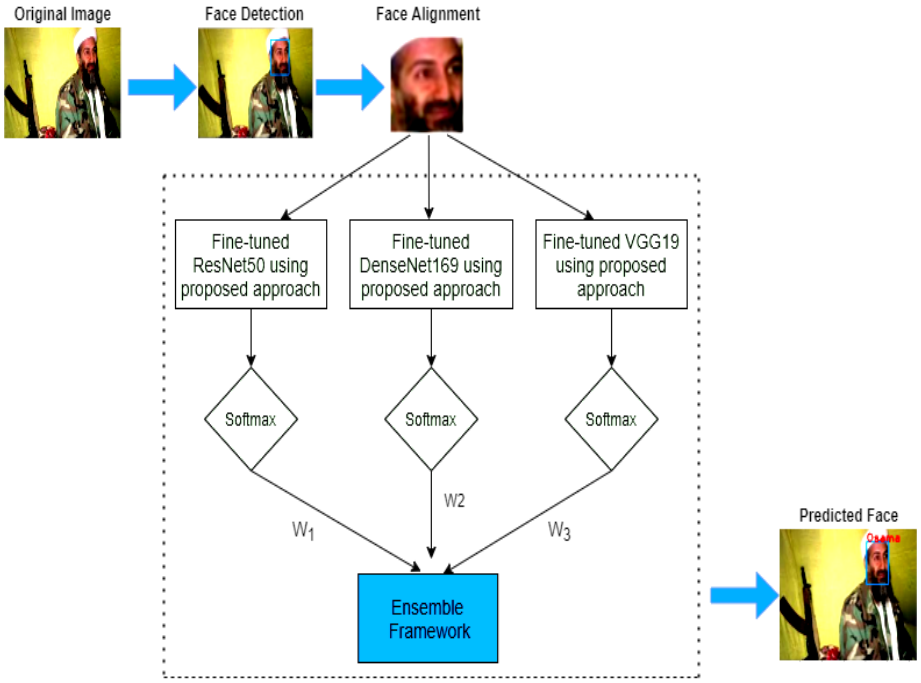


Figure 4.7 The Proposed Hybrid Ensemble CNN (HE-CNN) Model

4.4 Training of the Proposed Model and Hyperparameter Tweaking

In the present work, model training is done in two steps.

Step 1: First, freeze the early layers in the network and train only classification layers. However, the initial layers (*i.e.*, the feature extraction layers) are not trained during the first step of training the network.

Step 2: Then, a fine-tuned model is loaded from step 1, and unfreeze all the layers to train the complete model. Figure 4.8 provides a visual representation of the entire procedure.

The model's training process incorporates the utilization of a one-cycle policy [208], which replaces the conventional fixed learning rate with cyclical learning rates. The efficacy of hyperparameter tuning in enhancing machine learning model accuracy is well established. In the present approach, the adjustment of learning rate, batch size, image size, epochs, and dropout parameters is carried out

experimentally in Chapter 5 of the thesis to fine-tune the pre-trained models for the specific task at hand.

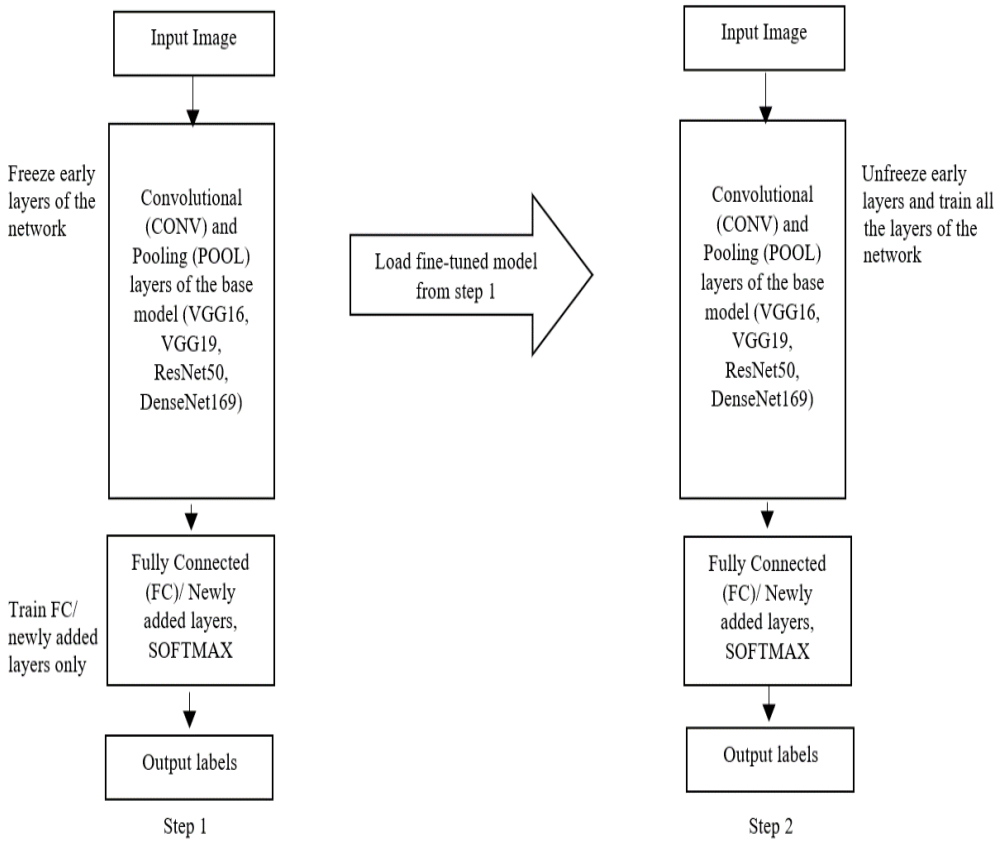


Figure 4.8 Steps for Training the Modified Models

The selection of an appropriate learning rate is of paramount importance. If the learning rate is overly large, it might lead to an overshoot of the optimal value; conversely, if it is excessively small, convergence to the optimal value may necessitate a protracted number of iterations. Thus, the Learning Rate Finder (LRF) curve [208] has been employed to ascertain the optimal learning rate for the model. The LRF curve is an invaluable tool for automatically identifying a suitable learning rate for any given model. For instance, as depicted by the red dot in Figure 4.9, it pinpoints the optimal learning rate for the deep learning model. The gradual elevation of the learning rate, initiated from an exponentially low value (*e.g.*, 10^{-6})

and progressing to a higher value (*e.g.*, 1), occurs during data training in small batches. The learning rate experiences oscillations between lower and upper boundaries during the cool-down phase before ultimately reverting to its initial low boundary. In the Softmax layer, a one-hot vector is made and used with categorical cross-entropy as the loss function to predict the type of data. This is shown in equation (4.29).

$$L(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C}\left[y_{ij}\log\left(p_{ij}\right)\right] \tag{4.29}$$

Where n is the count of images in the training data, i is the index of the input image (*i.e.*, i^{th} image), and j is the class's index, y_{ij} is the one-hot encoded label, and p_{ij} is the probability distribution over C classes. Adam [209] optimizer has been used for the minimization of loss function. It is a commonly used optimizer for deep learning models [210] [211] [212]. In this research work, the technique of early stopping [213] [214] has been implemented to counter overfitting and enhance the model's ability for generalization. This technique entails continuous monitoring of the model's performance on a validation set throughout the training phase, stopping the training process when the performance starts to deteriorate, and helping to achieve global minima. One indication of approaching the global minimum is the convergence of the loss function. If the loss decreases consistently during training and reaches a stable value, it suggests that the model is converging towards a good solution. However, it does not guarantee that the global minimum has been reached, as the model may still be trapped in a suboptimal solution. The complete procedure for the training process is illustrated in Algorithm 4.5. The algorithm comprises two functions: the first function is responsible for updating parameters during the training process when the initial layers are frozen, while the second function updates the weights of the entire model after introducing the proposed layers in the classification section by unfreezing all layers.

Algorithm 4.5 Algorithm of the Proposed Approach for Training to Obtain Fine-Tuned Models

Input: *Training Dataset* $D = \{x_i, y_i\}_{i=1}^{n}$, *where n is the count of input images in dataset, pre-trained CNN model (Here, VGG16, VGG19, ResNet50 and DenseNet169 are taken), no. of epochs (e), batch size (α), image size (m), calculated optimal learning rate using learning rate finder curve (η)*
Output: *Fine-tuned optimized CNN model for the addressed task (output_tuned_model)*

1. **procedure** *TL_step1 (D, CNN, e, α , m, η)*
2. //first train head and freeze remaining layers
   parameters ⟵ load_model(*CNN*, train_head=True)
3. **repeat**
4. **for all** $(x_i, y_i) \in D$ **do**
5. activation ⟵ forward_propagation(x_i, parameters)
6. cost ⟵ Loss_function(activation, y_i)
7. gradient ⟵ back_propagation(activation,cost)
8. parameters ⟵ weight_update(parameters, gradient, η)
9. **end for**
10. **until** e **times**
11. **return** ***tuned_model***
12. **end procedure**
13. **procedure** *TL_step2 (D, tuned_model, e, α, m, η)*
14. //Unfreeze all the remaining layers and train entire model
    //Load model received from function TL_step1 (*i.e.*, tuned_model)
    parameters ⟵ load_model(tuned_model, train_head=False)
15. **repeat**
16. **for all** $(x_i, y_i) \in D$ **do**
17. activation ⟵ forward_propagation(x_i, parameters)
18. cost ⟵ Loss_function(activation, y_i)
19. gradient ⟵ back_propagation(activation,cost)
20. parameters ⟵ weight_update(parameters, gradient, η)
21. **end for**
22. **until** e **times**
23. **return** ***output_tuned_model***
24. **end procedure**

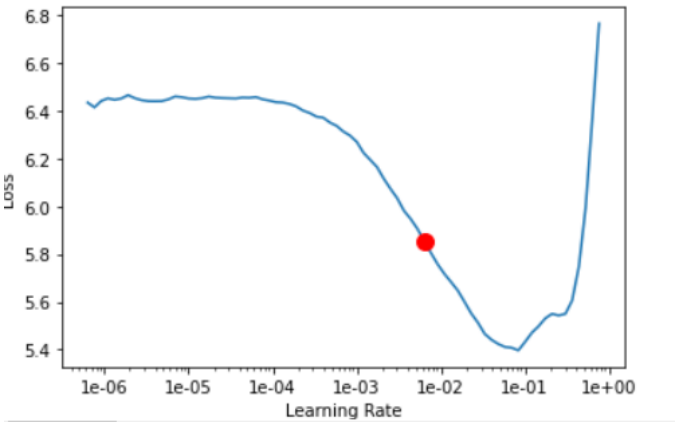


Figure 4.9 Learning Rate Finder Curve

4.4.1 IDENTIFICATION OF THE RANGE OF LEARNING RATE

The optimal learning rate has been calculated using a learning rate finder curve that can be automatically generated using the lr.find() function. The learning rate finder curve is a graph that shows the relationship between the learning rate and the corresponding loss or metric value during model training. It is typically plotted with the learning rate on the x-axis and the loss or metric value on the y-axis. While analyzing the plot given in Figure 4.10, it is important to note that the learning rate exponentially increases after each batch update. After completing a batch, the learning rate is increased for the next batch. Observing the plot, we can see that the loss remains relatively flat in the range of 1e-10 (*i.e.*, 10^{-10}) to 1e-6 (*i.e.*, 10^{-6}). This indicates that the learning rate is too small for the network to effectively learn any meaningful patterns. However, starting around 1e-5 (*i.e.*, 10^{-5}), the loss starts to decline, indicating that this is the minimum learning rate at which the network can actually learn. As the learning rate increases to approximately 1e-4 (*i.e.*, 10^{-4}), the network exhibits rapid learning. A slight increase in loss can be seen just past 1e-2 (*i.e.*, 10^{-2}), but the significant increase occurs at 1e-1 (*i.e.*, 10^{-1}). Towards the upper end, at 1e+1 (*i.e.*, 10), the loss dramatically escalates, indicating that the learning rate is excessively high for the model to learn properly. Based on this plot analysis,

we can visually determine the lower and upper bounds for the learning rate range, so that the lower bound is 1e-5 (*i.e.*, 10^{-5}) and the upper bound is 1e-2 (*i.e.*, 10^{-2}).

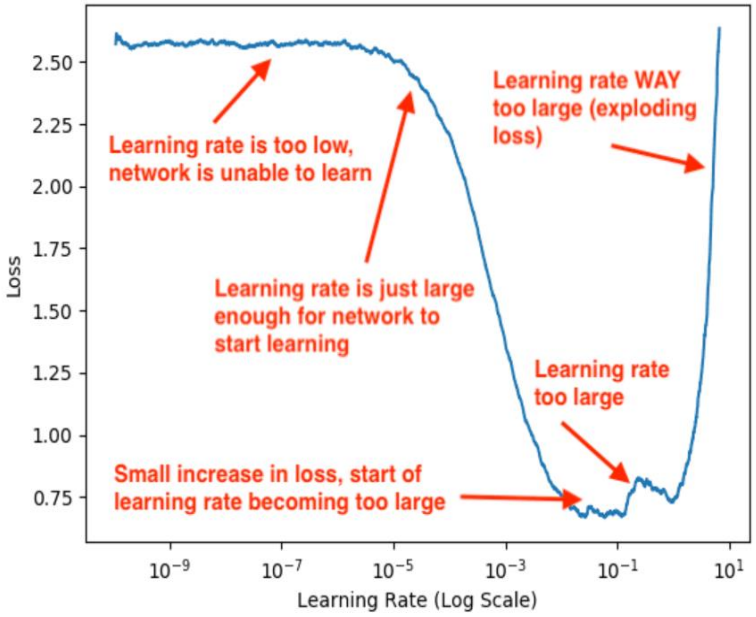


Figure 4.10 Identification of Learning Rate through LRF Curve [215]

4.5 Summary of the Chapter

This study centers on the implementation of an automated face recognition system comprising face detection, face recognition, alert generation, and the creation of clusters for regions with prominently identified desired faces. This proposed and implemented system has diverse applications, including locating missing individuals, criminal identification, and surveillance. In any face recognition process, the initial step entails detecting faces within images or videos, which our work accomplishes through SSD. For the face recognition stage, the present methodology employs transfer learning and ensemble learning to strike a balance between accuracy and computational efficiency.

The present approach encompasses two phases: In the first phase, we harness the ImageNet dataset to generate pre-trained weights. These weights are subsequently utilized in the second phase, wherein standard face datasets aid in generating high-level

features such as facial landmarks (eyes, nose, mouth, *etc.*) within images. The enhancement of the model's recognition accuracy has been significantly improved through the utilization of hyperparameter tuning, a fact substantiated through experimental evidence presented in the next chapter of the thesis. The alert generation phase has been executed using the Haversine formula, determining the proximity of an individual to the designated alert location. In the subsequent chapter of the thesis, the proposed system has been evaluated and conducted an ablation study to examine the effects of the suggested modifications.

CHAPTER-5

EXPERIMENTAL RESULTS AND DISCUSSION

Neurons serve as the computational units within a deep neural network (DNN), executing operations on data as it traverses the network. Each node within the DNN carries a weight value, learned during training, indicating its influence on prediction outcomes. These weights signify model parameters [216]. Hyperparameters, on the other hand, govern the training process. Crafting a Deep Neural Network (DNN) entails decisions like determining the hidden layer count between input and output layers and specifying node counts per layer. These aspects, while not directly tied to training data, are configuration variables. Hyperparameters typically remain fixed across tasks, while parameters change through training [216].

The endeavor to select optimal hyperparameter values for training a model using a tuned algorithm on a particular dataset is termed hyperparameter tuning. By optimizing model performance via a set of hyperparameters, the aim is to minimize a specified loss function, yielding improved results with fewer errors. Notably, the learning algorithm fine-tunes the loss based on input data, seeking the best solution within given constraints. Hyperparameters precisely shape this configuration. Inadequate adjustment of hyperparameters can lead to suboptimal outcomes even if model parameters are predicted accurately. In practice, the accuracy or confusion matrix might deteriorate [217].

For successful face recognition, meticulous hyperparameter tuning is crucial. The proposed model undergoes tuning via the judicious selection of pertinent parameters and hyperparameters for analysis and experimentation. This encompasses parameters like filter count, filter size, activation function, pooling size, *etc*., defined during model training at the sub-architecture level. It also involves decisions on learning strategies, weight initialization techniques to minimize the cost function, and hyperparameters such as learning rate, batch size,

image size, epoch count, and dropout, as discussed in the preceding chapter. The impact of these parameter and hyperparameter choices on face recognition accuracy is experimentally demonstrated in this chapter, aiming to illustrate the achievement of optimal outcomes.

5.1 Experimental Setup and Evaluation Parameters

Experiments have been conducted to independently assess the efficacy of face detection and recognition models. After assessing and comparing its performance with various available algorithms in Section 5.2, SSD is chosen to identify and align the faces for the recognition phase. The system is set up with Windows 10 and 16GB of RAM, an NVIDIA GTX 1650 Ti with a 4GB GPU, and an AMD Ryzen 5 4600H with Radeon Graphics. TensorFlow version is 2.4.0, whereas Keras and OpenCV versions are 2.4.3 and 4.5.1, respectively. Keras is used for the detection phase, and the implementation of the recognition stage is done using Fastai version 1.0.61. The records of mugshots and police officers are stored in the SQLite database.

The present research considered three evaluation parameters for the assessment of face detection algorithms: True Positive Rate (TPR), False Negative Rate (FNR), and False Positive Rate (FPR). TPR can also be called recall or sensitivity. It is the ability of the classification model to identify all the significant instances. FPR is the total count of false-negative assessments divided by the number of all negative evaluations. FNR shows the proportion of correct results that were missed and classified as incorrect. The formulae to estimate the values of TPR or recall, FPR, and FNR are given in equations (5.1) – (5.3), where TP refers to having both the actual and predicted label the same. For example, an image contains a face, and the algorithm also detects it as a face. FP is defined as the true label not being a face, but the predicted label being a face [218]. FN has the true label as a face, but the predicted label does not have a face. The definitions of the mentioned measures are depicted in Figure 5.1.

$$TPR\ or\ Recall = \frac{TP}{TP+FN} \tag{5.1}$$

$$FPR = \frac{FP}{FP+TN} \tag{5.2}$$

$$FNR = \frac{FN}{FN+TP} \tag{5.3}$$

The classification accuracy is the evaluation parameter used to calculate the performance of the discussed fine-tuned modified models and other SOTA for the three considered datasets, which is calculated using the formula given in equation (5.4).

$$Accuracy = \frac{Number\ of\ correctly\ recognized\ images}{Total\ number\ of\ images} \tag{5.4}$$

The other two evaluation measures, such as precision and recall, have been used to assess the classification model because accuracy alone is insufficient to choose the best classifier due to the accuracy paradox [219]. Precision defines the number of true positives out of the predicted positives. Recall and precision can be derived from the formulas given in equations (5.1) and (5.5). The Receiver Operating Characteristic (ROC) curve is also used to evaluate the performance of the proposed modified models. The ROC curve gives an estimation of the rate of true positives relative to the rate of false positives for the classifier. In other words, it highlights the sensitivity of the classifier [220]. In addition, the total number of inaccurate predictions on the test set divided by all of the test set predictions can be used to compute the error rate given in equation (5.6). We can always determine accuracy from the error rate since they are complementary quantities.

$$Precision = \frac{Truly\ Positives}{Predicted\ Positives} \tag{5.5}$$

$$Error\ Rate = \frac{Number\ of\ incorrect\ predictions}{Total\ number\ of\ images} \tag{5.6}$$

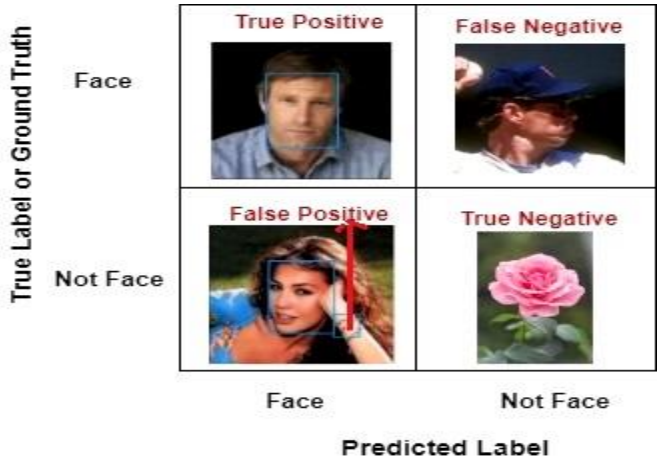


Figure 5.1 Confusion Matrix for Face Detection

5.2 Results and Discussion on Various Modules of the Proposed Face Recognition System

5.2.1 Self-Curated Dataset and Database of Criminals' and Police Officials' Records

In the first module, we collected images of criminals from the Internet (freely available sources) and stored those images in different directories labeled with their names, as given in Figure 5.2. The information about the mugshots, such as the crime date, crime type, age, *etc.*, is stored in the database, as delineated in Figure 5.3 (a). The other table in the database is also created to store the registered email ID, mobile number, and location coordinates of the police stations manifested in Figure 5.3 (b). Personal numbers are used for testing the system; that is why they are scraped in the image given in Figure 5.3 (b) due to privacy concerns.

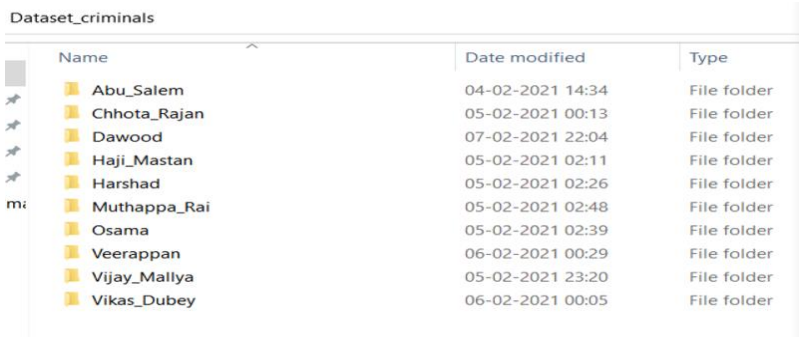


Figure 5.2 Images of Criminals Collected from the Internet

Criminal_ID	Name	Age	Gender	Crime_type	Crime_date	Identity_mark
Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	Abu_Salem	59	M	Robbery;murder;gangster	2019-01-01;2020-05-01;2020-12-03	Cut mark on forehead
2	Chhota_Rajan	62	M	Mobster;smuggling;extortion	2019-02-01;2020-05-01;2020-12-03	Cut mark on left hand
3	Dawood	44	M	Murder;gangster;terrorism	2018-02-01;2018-01-01;2020-12-03	Cut mark on right eye
4	Haji_Mastan	68	M	Kidnapping;murder	2018-02-01;2019-02-01	Cut mark on left eye
5	Harshad	47	M	Scam	2018-02-01	Cut mark on left elbow
6	Muthappa_Rai	42	M	Smuggling, Kidnapping	2018-02-01;2019-02-01	Cut mark on right elbow
7	Osama	55	M	Terrorism;bombblast	2018-02-01;2019-02-01	Cut mark on forehead
8	Veerappan	50	M	Kidnapping;robbery	2018-02-01;2019-02-01	Cut mark on right hand
9	Vijay_Mallya	65	M	Scam	2018-02-01	Cut mark on chest
10	Vikas_Dubey	40	M	Stolen property;kidnapping;murder	2018-02-01;2019-02-01;2020-02-05	Cut mark on chin

(a)

Police_Chowki_ID	Police_chowki_Name	Email	Phone_number	latitude	longitude
Filter	Filter	Filter	Filter	Filter	Filter
1	Laxman Chowk Police Chowki	sanwarul@ddn.upes.ac.in	[redacted]	30.321661	78.021585
2	Police Sahayata Kendre	dgc-police-ua@nic.in	9411112780	30.338329	78.020920
3	Bindaal Police Chowki	dgc-police-ua@nic.in	1352716235	30.329576	78.031539
4	Lakhibagh Police Station	dgc-police-ua@nic.in	9411112809	30.316359	78.032544
5	Kotwali Police Station	dgc-police-ua@nic.in	1352716216	30.320393	78.037553
6	Police Headquarters SSP Office ...	ssp-deh-ua@nic.in	1352716203	30.317734	78.039864
7	Police Chowki Khudura	dgc-police-ua@nic.in	1352616216	30.321719	78.032672
8	Patel Nagar Police Station	dgc-police-ua@nic.in	1352716219	30.292877	78.017799
9	Dhara Chowki	dgc-police-ua@nic.in	1352716216	30.325680	78.042870
10	Uttarakhand Police Headquarters	dgc-police-ua@nic.in	1352712685	30.329997	78.050405
11	Panditwari Police Chowki	sanwarul@ddn.upes.ac.in	[redacted]	30.3324	77.9881
12	Prem nagar Police Chowki	shahinaanwarul@gmail.com	[redacted]	30.333092	77.961016

(b)

Figure 5.3 Record Stored in Database (a) Criminals' Records (b) Police Officials' Records

5.2.2 Detection and Recognition Module

This module contains two steps, namely face detection and face recognition. The first step is to detect the faces and then compare the detected and aligned faces from the gallery images to recognize the mugshot.

5.2.2.1 Face Detection

Face detection is a technology that identifies human faces in an image. Various traditional and deep learning-based methods have been introduced as SOTA for face detection. Traditional approaches like the Haar classifier, also known as the Viola-Jones algorithm, and the LBP classifier for face detection have their benefits and drawbacks, but the major differences are in terms of speed and accuracy. So, a Haar classifier is used in cases where there is a requirement for more accurate detections. But the LBP classifier is faster and, therefore, should be used in mobile applications or embedded systems [221]. The Haar classifier and the LBP classifier, both conventional algorithms, fail to achieve the desired detection accuracy, as demonstrated by the experimental findings presented in Table 5.1 and Table 5.2. Deep network approaches like MTCNN and SSD for face detection are efficacious in terms of their detection accuracy.

Table 5.1 Detection Accuracy (Number of Detected Faces/Total Faces in an Image) and Time (in Sec) of Face Detection Algorithms on Sample Images

S. No.	Face Detection Algorithm	No. of Faces Found in an Image/ No. of Faces Present in an Image	Time (in Sec)
1.	SSD (Single Shot Multi-Box Detector)	Image 1: 2/2	**0.051**
		Image 2: 5/5	**0.032**
2.	MTCNN (Multi-Task Cascaded Convolutional Networks)	Image 1: 2/2	2.723
		Image 2: 5/5	2.998
3.	Haar Cascade (Viola Jones)	Image 1: 1/2	0.234
		Image 2: 4/5	0.051
4.	LBP Cascade	Image 1: 1/2	0.112
		Image 2: 1/5	0.045

Figure 5.4 Outputs of Different Face Detection Algorithms: (a) Face Detection using Haar Cascade (b) Face Detection using LBP Cascade (c) Face Detection using MTCNN (d) Face Detection using SSD

Images containing multiple faces are considered for the evaluation of the State-of-the-Art face detection algorithms because a video frame can have more than one face at a particular instant in time. From Table 5.1, Figure 5.4 , and Table 5.2, it is experimentally proven that Haar cascade is accurate but slower than LBP cascade, while LBP is faster but less accurate. MTCNN and SSD are used in applications where accuracy has greater importance. But SSD is much faster in comparison to MTCNN and provides equivalent accuracy. The evaluation of the discussed face detection methods is also conducted using standard datasets like LFW, CPLFW, and the self-curated dataset in Table 5.2. After evaluating all the discussed methods, it is concluded that SSD is an efficient approach in terms of accuracy and

processing time for face detection. Therefore, we used the SSD framework to detect faces for the face recognition stage in the proposed recognition system.

Table 5.2 Detection Score of Various Face Detection Algorithms (in %)

		Face Detection Algorithm											
		SSD			**MTCNN**			**LBP Cascade**			**Haar Cascade**		
S. No.	**Dataset**	**TPR**	**FPR**	**FNR**	**TPR**	**FPR**	**FNR**	**TPR**	**FPR**	**FNR**	**TPR**	**FPR**	**FNR**
1.	CPLFW	96.9	1.4	1.7	96.3	1.3	2.4	44.2	0.7	55.1	53.9	0.5	45.6
2.	LFW	99.2	0.8	0	99.3	0.7	0	94.3	0.8	4.9	98.4	0.6	1
3.	Criminal Dataset	92.9	1.7	5.4	93.4	2.3	4.3	44	0.6	55.4	53.7	1.3	45

5.2.2.2 Face Recognition

Experimental Results on LFW Dataset

The split ratio of the dataset used in the experiment is 7:3, *i.e.*, 70% of the samples in the dataset have been used for training, and 30% have been used for validation. Chapter 3 of the thesis provides an in-depth overview of the dataset. Classes containing more than one sample are considered for experimental evaluation (*i.e.*, 1680 classes are considered). Data augmentation [114] [222], known as oversampling, has been utilized through a series of standard transformations such as vertical flip, horizontal flip, rotation, zooming, warping, scaling, and lighting to ensure balance in the considered classes. The total number of considered images for experimental evaluation is 13,440 after data augmentation (*i.e.*, 9,408 are used for training and 4,032 are used for validation). Dropout values are set to 0.1 and 0.2 for the layers used in the modified architecture after rigorous

experiments. Training and validation losses for different models are illustrated in Table 5.3, and Table 5.4 lists the comparison of the intended approach with other existing techniques. The learning rate has been chosen randomly for pre-trained models, while modified pre-trained models utilize the learning rate identified through the learning rate finder curve. Figures 5.5 and 5.6 illustrate the training and validation losses over batches processed of pre-trained and modified pre-trained models, respectively. In Tables 5.3, 5.5, and 5.7, Train head=T (True) means training of the head only and the remaining layers are freezed, while Train head=F (False) denotes that all the layers are unfreezed and training is done for the complete model. The ROC curves for the pre-trained models and modified models are shown in Figure 5.7.

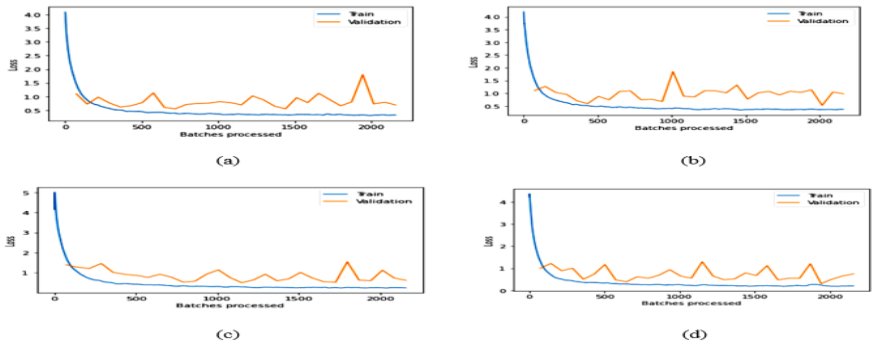


Figure 5.5 Training and Validation Loss over Batches Processed Graphs for Pre-Trained (a) VGG16, (b) VGG19, (c) ResNet50, and (d) DenseNet169 in the LFW Dataset

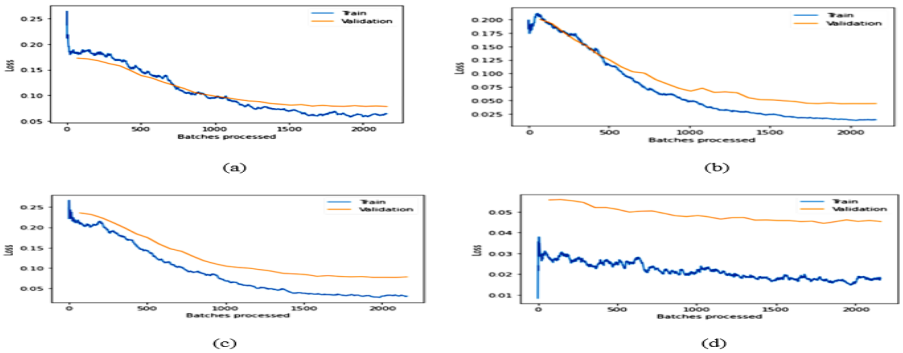


Figure 5.6 Training and Validation Loss over Batches Processed Graphs for Modified (a) VGG16, (b) VGG19, (c) ResNet50, and (d) DenseNet169 in the LFW Dataset

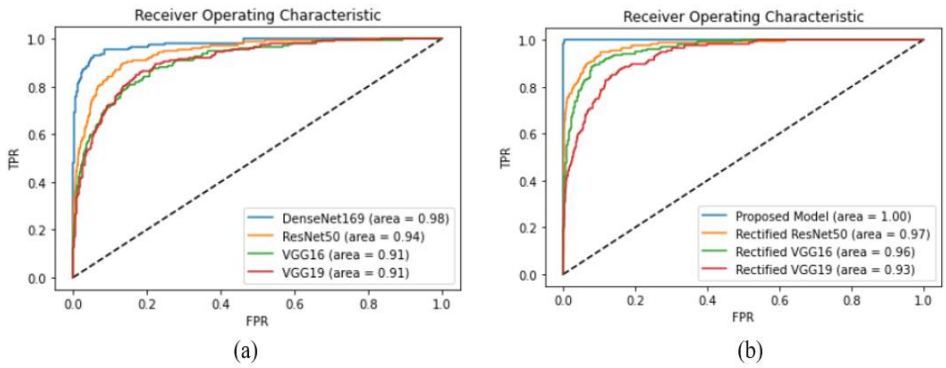


Figure 5.7 ROC Curves on LFW Dataset: (a) ROC Curves of Pre-Trained Models (b) ROC Curves of Modified Models

Table 5.3 Training and Validation Loss of Pre-Trained Models (VGG16, VGG19, ResNet50, and DenseNet169) and Modified Pre-Trained Models with Proposed Classifier (PC) in LFW

S. No.	MU	TH= T/F	E	LR	TL	VL	ER	TA	VA	P	R
	Pre-Trained Models										
1.	VGG16	F	30	1e-2	0.323	0.690	0.201	0.925	0.798	0.826	0.796
2.	VGG19	F	30	1e-2	0.369	0.981	0.271	0.917	0.728	0.828	0.725
3.	ResNet 50	F	30	1e-2	0.250	0.623	0.169	0.922	0.830	0.860	0.832
4.	DenseNet 169	F	30	1e-2	0.210	0.752	0.188	0.923	0.811	0.862	0.812
	Modified Pre-Trained Models with Proposed Classifier (PC)										
5.	Modified VGG16	T	15	1e-2	0.193	0.173	0.051	0.934	0.948	0.949	0.950
		F	30	1e-6, 8e-5	**0.064**	**0.078**	**0.022**	**0.981**	**0.977**	**0.977**	**0.977**

6.	Modified VGG19	T	15	1e-2	0.211	0.204	0.064	0.933	0.935	0.935	0.935
		F	30	1.1e-5, 1e-4	**0.014**	**0.043**	**0.013**	**0.991**	**0.986**	**0.986**	**0.986**
7.	Modified ResNet50	T	15	2.09e-3	0.209	0.241	0.075	0.931	0.924	0.924	0.926
		F	30	1.91e-6, 8e-5	**0.031**	**0.078**	**0.023**	**0.990**	**0.976**	**0.976**	**0.976**
8.	Modified DenseNet169	T	15	1e-2	0.028	0.056	0.015	0.996	0.984	0.984	0.984
		F	30	1.32e-6, 1e-5	**0.018**	**0.045**	**0.011**	**0.998**	**0.988**	**0.988**	**0.988**

(*Train_head is represented as TH, model used as MU, epoch as E, learning rate as LR, training loss as TL, validation loss as VL, error rate as ER, training accuracy as TA, validation accuracy as VA, precision as P, and recall as R)

Table 5.4 The Comparison of the Proposed Work with other SOTA in the LFW Dataset

S. No.	Author, Year of Publication	Techniques Used	Accuracy (%)	Error rate (%)
1.	Proposed work	The hybrid model of the fine-tuned pre-trained models using ensemble learning (HE-CNN)	**99.35**	**0.65**
2.	Mishra *et al.* [223], 2022	Deep learning architectures + Hardmining loss	95.55	4.45
3.	Ben Fredj *et al.* [224], 2021	GoogleNet +Data augmentation	99.20	0.80
4.	Kang [225], 2019	Self-learning CNN	94.9	5.10
5.	Wen *et al.* [113], 2016	Combination of softmax loss and center loss with CNN	99.28	0.72
6.	Parkhi *et al.* [98], 2015	Deep CNN	98.95	1.05
7.	Sun *et al.* [27], 2014	DeepID	97.45	2.55

Experimental Results on CPLFW Dataset

In this experimental evaluation, random splitting of the CPLFW dataset has been done with a splitting ratio of 8:2 (*i.e.*, 80% of images (9,322) have been used

for training and 20% (2,330) have been used for validating the model). The dropout value is set between 0.25 and 0.5. Training and validation losses for different models are mentioned in Table 5.5. Data augmentation is not applied to the images of the dataset as it contains balanced classes. The comparison of the proposed approach with other existing techniques is listed in Table 5.6. Finally, the training and validation loss over batches processed graphs for the pre-trained and fine-tuned baseline models are delineated in Figures 5.8 and 5.9. The ROC curves for the pre-trained models and modified models are shown in Figure 5.10.

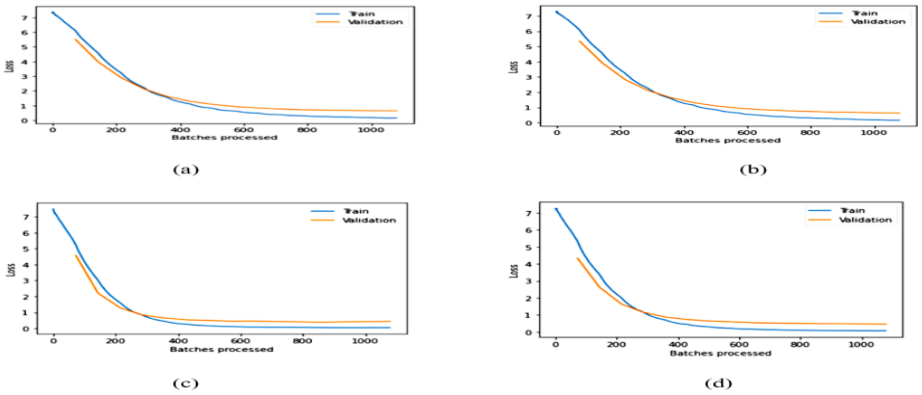


Figure 5.8 Training and Validation Loss over Batches Processed Graphs for Pre-Trained (a) VGG16, (b) VGG19, (c) ResNet50, and (d) DenseNet169 in the CPLFW Dataset

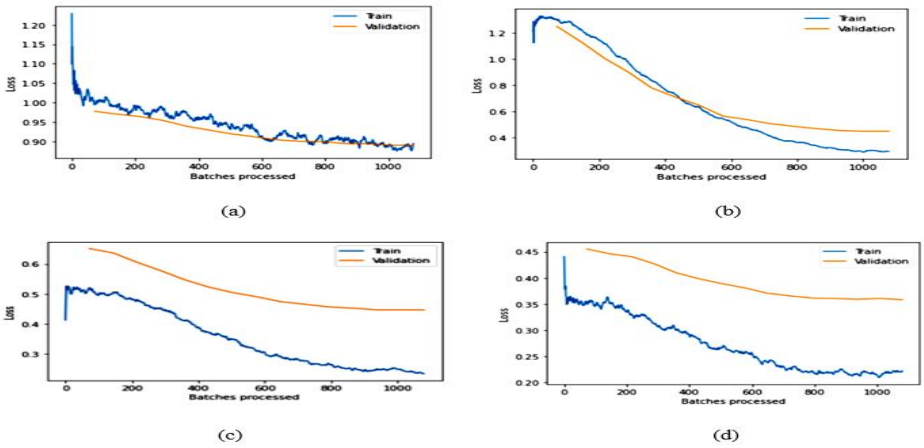


Figure 5.9 Training and Validation Loss over Batches Processed Graphs for Modified (a) VGG16, (b) VGG19, (c) ResNet50, and (d) DenseNet169 in the CPLFW Dataset

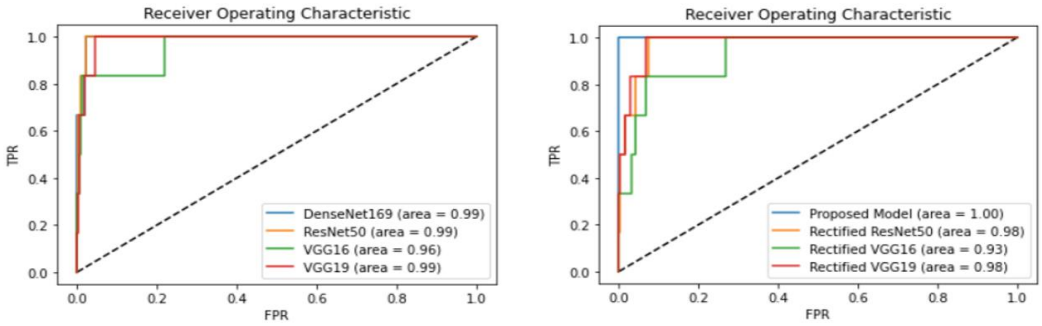


Figure 5.10 ROC Curves on CPLFW Dataset: (a) ROC Curves of Pre-Trained Models (b) ROC Curves of Modified Models

Table 5.5 Training and Validation Loss of Pre-Trained Models (VGG16, VGG19, ResNet50, and DenseNet169) and Modified Pre-Trained Models with Proposed Classifier (PC) in CPLFW

S. No.	MU	TH= T/F	E	LR	TL	VL	ER	TA	VA	P	R
	Pre-Trained Models										
1.	VGG16	F	15	1e-3	0.158	0.641	0.121	0.963	0.878	0.871	0.873
2.	VGG19	F	15	1e-3	0.162	0.635	0.128	0.956	0.871	0.865	0.872
3.	ResNet 50	F	15	1e-3	0.067	0.450	0.090	0.985	0.909	0.900	0.905
4.	DenseNet 169	F	15	1e-3	0.050	0.437	0.088	0.989	0.911	0.920	0.919
	Modified Pre-Trained Models with Proposed Classifier (PC)										
5.	Modified VGG16	T	7	2.75e-2	1.162	0.975	0.207	0.773	0.792	0.779	0.798
		F	15	6.31e-7, 1e-5	**0.894**	**0.889**	**0.189**	**0.811**	**0.810**	**0.826**	**0.816**

6.	Modified VGG19	T	7	1.1e-2	1.439	1.292	0.260	0.724	0.739	0.725	0.734
		F	15	1e-5, 5e-4	**0.292**	**0.446**	**0.084**	**0.932**	**0.915**	**0.916**	**0.913**
7.	Modified ResNet50	T	7	1.32e-2	0.594	0.658	0.131	0.886	0.868	0.876	0.867
		F	15	2.2e-6, 1e-4	**0.233**	**0.446**	**0.084**	**0.945**	**0.915**	**0.924**	**0.915**
8.	Modified DenseNet169	T	7	1.32e-2	0.433	0.456	0.087	0.919	0.912	0.913	0.913
		F	15	5e-6, 5e-5	**0.222**	**0.358**	**0.084**	**0.928**	**0.915**	**0.923**	**0.921**

(*Train_head is represented as TH, model used as MU, epoch as E, learning rate as LR, training loss as TL, validation loss as VL, error rate as ER, training accuracy as TA, validation accuracy as VA, precision as P, and recall as R)

Table 5.6 The Comparison of the Proposed Work with other SOTA in the CPLFW Dataset

S. No.	**Author, Year of Publication**	**Techniques Used**	**Accuracy (%)**	**Error rate (%)**
1.	Proposed work	Hybrid model of the fine-tuned pre-trained models using ensemble learning	**91.58**	**8.42**
2.	Liu *et al.* [226], 2021	Lightweight CNN	89.52	10.48
3.	Cao *et al.* [28], 2018	VGGFace2	84.10	15.90
4.	Liu *et al.* [227], 2017	SphereFace	81.40	18.60

Based on the experimental findings regarding facial recognition algorithms on the LFW and CPLFW datasets, relying solely on pre-trained models is inadequate for achieving optimal accuracy. Some modifications need to be implemented to improve the recognition accuracy of the models. After obtaining fine-tuned modified baseline models, ensemble learning can be utilized to get SOTA competent recognition rates in standard datasets, as illustrated in Tables 5.4 and 5.6. The original pre-trained models are trained on ImageNet, comprising 1000

classes. However, the number of classes mentioned in the last fully connected layer of the pre-trained models is insignificant in our experiments. Therefore, the approach in the presented work modified only the last fully connected layer to mention the count of classes in the used datasets for the experimental evaluation. The above results and graphs show that the pre-trained models, without any modification, do not provide the desired results. The spikes in the graphs of pre-trained models in LFW show unstable validation accuracy during the whole training process, and accuracy is also significantly less than the proposed approach. The reason for getting too many spikes in validation loss can be the large value of the learning rate resolved in the graphs of CPLFW by taking the small value of the learning rate. Therefore, using the learning rate finder curve to identify the optimal learning rate for the model instead of taking random values is suggested. If the training loss keeps decreasing while the validation loss increases or remains constant, this is a sign of overfitting. It is evident from Figures 5.5 and 5.8 that data overfitting is alleviated by the suggested approach, as shown in Figures 5.6 and 5.9. The results in Tables 5.3 and 5.5 show that the present approach gives better results for VGG19, Resnet50, and DenseNet169; that is why these three fine-tuned models are considered for designing the ensemble model (HE-CNN). The proposed modifications in the classification layer and training process generated SOTA-competent results and improved the recognition accuracy of the pre-trained models in LFW up to approximately 26% and 4% in CPLFW. The proposed and implemented ensemble model achieved competent accuracy compared to other existing methods requiring millions of identities to train the network (*i.e.*, high GPU memory consumption and computational cost are required for existing methods like ArcFace, FaceNet, *etc.*). The ROC curves of the modified models given in Figures 5.7 and 5.10 show the good fit of the modified models.

Experimental Results on GT Face Dataset

The validation accuracy of the GT face dataset is approaching 100% recognition accuracy on a very small number of epochs for the proposed model. The dataset does not contain a variation of unconstrained factors. All the faces are frontal, and a very low illumination effect is considered. Therefore, it can be concluded that high accuracy is achieved with minimal effort if images are taken in a constrained environment. In the first phase of training, the image size is 128x128 and the batch size is 120, while in the second phase, the image size is 224x224 and the batch size is 96. The variation in image and batch sizes makes the model robust against the size of the image. The best model for the GT face database is achieved by considering the split ratio (SR) of 8:2, oversampling (O) using standard transformations, dropout (D) values of 0.12 and 0.25, image size (IS) of 224x224, training of all the layers (TH=F), optimal cyclic learning rates (LR), and 20 epoch (E) size. The model's performance is evaluated using training loss (TL), validation loss (VL), training accuracy (TA), validation accuracy (VA), precision (P), and recall (R). The difference in training and validation losses shown in Table 5.7 and Figure 5.11 demonstrates the reduction of underfitting in the model. Data augmentation has been used to increase the number of images in each class of the dataset. Each class is oversampled by a series of transformations, and a total of 2500 images (50 samples per class) have been considered for the experiments. Table 5.8 illustrates the comparison of the modified model with other existing SOTA approaches.

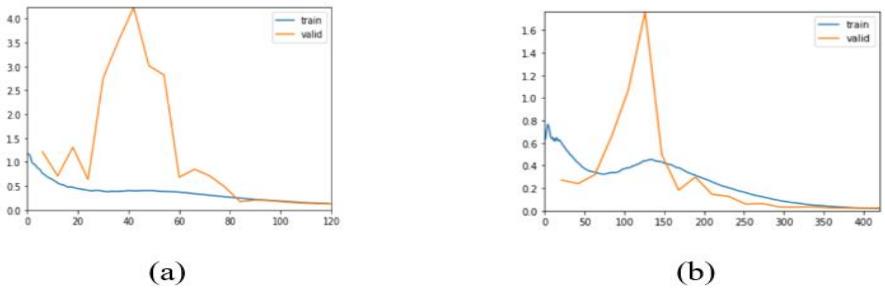


Figure 5.11 Training and Validation Loss vs Processed Batches Curve (a) Without Oversampled Dataset (b) With Oversampled Dataset.

Table 5.7 Experimental Results of GT Face Dataset

S. No.	SR	TH= T/F	D	E	LR	TL	VL	TA	VA	P	R
	Oversampling										
1.	7:3	T	0.25, 0.5	5	1e-1	2.115	0.683	0.723	0.808	0.821	0.812
2.		F		10	1e-3, 3e-3	0.055	0.016	0.982	0.991	0.991	0.990
3.	7:3	T	0.12, 0.25	10	1e-2	1.229	0.609	0.798	0.844	0.847	0.842
4.		F		20	1e-3, 5e-3	0.048	0.017	0.991	0.993	0.993	0.992
5.	8:2	T	0.25, 0.5	10	6e-2	0.762	0.276	0.898	0.927	0.926	0.917
6.		F		20	5e-4, 2e-3	0.020	0.024	0.994	0.992	0.993	0.991
7.	**8:2**	T	**0.12, 0.25**	10	6e-2	0.512	0.229	0.910	0.940	0.949	0.940
8.		**F**		**20**	**6e-4, 1e-3**	**0.014**	**0.023**	**0.998**	**0.996**	**0.997**	**0.995**
	Without Oversampling										
9.	8:2	T	0.12, 0.25	10	6e-2	0.132	0.411	0.814	0.807	0.801	0.799
10.		F		20	5e-4, 2e-3	0.130	0.123	0.981	0.983	0.981	0.980

Table 5.8 The Comparison of the Proposed Work with other SOTA in the GT Face Dataset

S. No.	Author, Year of Publication	Techniques Used	Accuracy (%)	Error rate (%)
1.	Proposed work	Hybrid model of the fine-tuned pre-trained models using ensemble learning	**99.63**	**0.37**
2.	Zhang *et al.* [228], 2022	Dictionary learning	76.67	23.33
3.	Muqeet *et al.* [229], 2019	LBP based on directional wavelet transform	82.25	17.75
4.	Ayyad *et al.* [230], 2019	SVD+LDA	89.24	10.76
5.	Dora *et al.* [231], 2017	Gabor filter + Minimum Distance Classifier (MDC)	92.50	7.50

Experimental results on YTF dataset

The V2V recognition is achieved with approximately 100% accuracy as a large number of video frames of an individual are used for training (*i.e.*, the training set is large). The experimental results on the YTF dataset have been done to show the recognition accuracy of the V2V face recognition approach. The confusion matrix and the output of the hybrid model on the YTF dataset are illustrated in Figures 5.12 and 5.13, respectively. The number of frames in each class of dataset is varied. Therefore, a total of 4,587 frames from 50 individuals were used for the experimental results. The recognition accuracy of the hybrid model for the YTF dataset is achieved by considering the split ratio (SR) of 8:2, dropout (D) values of 0.25 and 0.5, frame size (FS) of 224*224, optimal cyclic learning rates (LR), and 20 epoch (E) size for all three considered modified pre-trained models. The diagonal values in the confusion matrix, as given in Figure 5.12, represent the number of correctly predicted faces. The output of the face recognition model is illustrated in Figure 5.13 in the form of the actual class/ predicted class/ loss/ probability of the predicted class. The comparison of the present work with other existing SOTA approaches is given in Table 5.9, which shows the SOTA competence of the presented work.

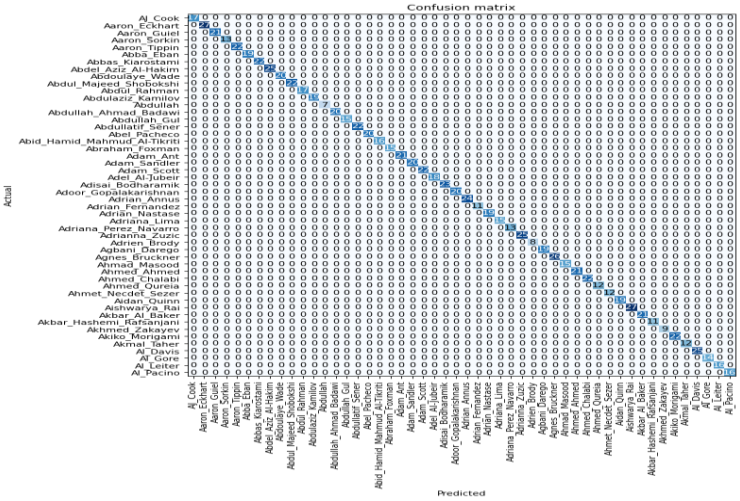


Figure 5.12 Confusion Matrix of YTF Dataset Results

Figure 5.13 Output of the Face Recognition Model on YTF Dataset

Table 5.9 The Comparison of the Proposed Work with other SOTA in the YTF Face Dataset

S. No.	Author, Year of Publication	Techniques Used	Accuracy (%)	Error rate (%)
1.	Proposed work	Hybrid model of the fine-tuned pre-trained models using ensemble learning	**99.21**	**0.79**
2.	Ben Fredj *et al.* [224], 2021	GoogleNet +Data augmentation	96.60	3.40
3.	Liu *et al.* [232], 2021	EQFace	98.18	1.82
4.	Ding *et al.* [132], 2017	TBE-CNN	94.96	5.04
5.	Schroff *et al.* [26], 2015	FaceNet	95.10	4.90
6.	Taigman *et al.* [31], 2014	DeepFace	91.40	8.60

Experimental Results on Self-Curated Criminal Dataset

A small criminal dataset has been created containing 25 images of each class of criminals, namely Haji Mastan, Vijay Mallya, Dawood, Harshad, Osama, Veerappan, Chhota Rajan, Muthappa Rai, Abu Salem, and Vikas Dubey, by downloading these images from the Internet to demonstrate the real-time application of face recognition. Mislabeled and vague images from the downloaded images are manually deleted, and 25 images of each class are considered to make a class-balanced dataset. Data augmentation, known as the oversampling technique, has been utilized to expand the count of samples in each class. In order to maintain the balance of the class, images in individual classes have been augmented to generate 50 samples using a set of transformations such as vertical flip, horizontal flip, mirroring, warping, scaling, rotation, zooming, and lighting. The dataset is divided into two sets consisting of 80% and 20% samples (*i.e.*, 400 samples of the dataset are considered for training and 100 samples are taken for testing). Testing on a self-created dataset has been done in two ways. Firstly, the testing has been done using 100 random samples from 500 images. The accuracy of the presented hybrid model on a self-created dataset is 95%, while the precision score, recall score, and error rate are 0.954, 0.952, and 5%, respectively. The confusion matrix to analyze the correct results present in the diagonal of the matrix is delineated in Figure 5.14.

Secondly, another testing dataset contains 50 images with more than one face to demonstrate the real-time surveillance results. The set of 50 images is distinct from the collection of 500 images but consists of the faces of the same 10 criminals and other unknown individuals. As the testing images contain more than one face, the recognition rate of the proposed technique in the criminal dataset has been calculated by manually analyzing each image, as shown in Figure 5.15, and achieving 87% recognition accuracy.

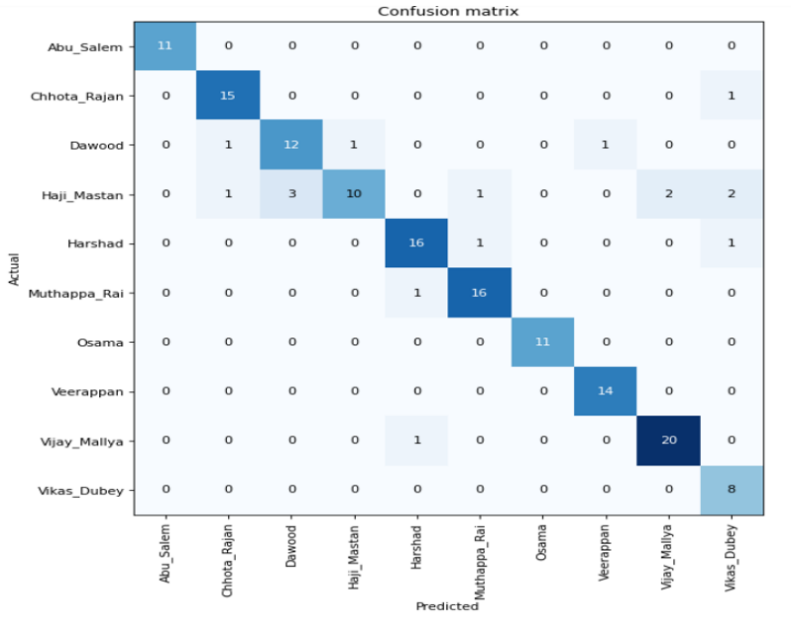


Figure 5.14 Confusion Matrix of Self-Curated Dataset Results

Figure 5.15 Output of the Face Recognition Stage on Self-Curated Dataset

5.2.2.3 Ablation Evaluation of the Modified Baseline Models for Face Recognition

An ablation study is a research methodology commonly employed in machine learning and experimental sciences to investigate the impact of individual components or factors within a complex system, such as a machine learning model.

The term "ablation" refers to the removal or alteration of specific components or elements within the system to assess their contribution to overall system performance. In the context of machine learning models, an ablation study involves systematically disabling or modifying specific parts of the model or its training process to analyze their influence on the model's performance. By conducting ablation studies, researchers can fine-tune machine learning models, gain a deeper understanding of their inner workings, and make informed decisions about which components or methods are crucial for achieving the desired results. This methodology contributes to the advancement and optimization of machine learning algorithms and models [233] [234]. The study has been done to highlight the impact of various modifications made to the presented work on recognition accuracy, as illustrated in Table 5.10. Here, the LFW dataset has been utilized to perform the ablation evaluation. The configuration for the experiments is the same as discussed in Section 5.1.

Table 5.10 Ablation Study of Modified Baseline Models

S. No.	Components for Ablation Evaluation	RA (in %) of Modified DenseNet169	RA (in %) of Modified ResNet50	RA (in %) of Modified VGG19
1.	**Effect of two-phase learning**			
	• Recognition accuracy with two-phase learning	98.84	97.61	98.64
	• Recognition accuracy without two-phase learning	98.49	92.49	93.55
2.	**Effect of dropout layer**			
	• Recognition accuracy with dropout layer	98.84	97.61	98.64
	• Recognition accuracy without dropout layer	90.10	92.53	93.82
3.	**Effect of learning rate**			
	• Recognition accuracy with fixed learning rate	81.13	83.07	72.83
	• Recognition accuracy with cyclical learning rate	98.84	97.61	98.64
4.	**Effect of concatenation of GAP and GMP**			

	• Recognition accuracy with concatenation of GAP and GMP	98.84	97.61	98.64
	• Recognition accuracy without concatenation (only average pooling)	98.20	96.23	98.10
5.	**Effect of optimizer**			
	• Recognition accuracy with Adam optimizer	98.84	97.61	98.64
	• Recognition accuracy with SGD optimizer	98.10	96.23	97.81
6.	**Effect of activation function**			
	• Recognition accuracy with ReLU	97.90	96.53	97.32
	• Recognition accuracy with Leaky ReLU	98.84	97.61	98.64
7.	**Effect of the addition of fully connected layer**			
	• Recognition accuracy with one/ three fully connected layer	97.10	96.34	98.52
	• Recognition accuracy with two fully connected layer	98.84	97.61	98.64

(*RA= Recognition Accuracy)

a) Effect of two-phase learning on recognition accuracy: The concept of a two-phase training process discussed in Chapter 4 of the thesis has been adopted in the present approach for training the model. This approach significantly increases recognition accuracy, as shown in Table 5.10.

b) Effect of the dropout layer on recognition accuracy: The use of the dropout layer in the modified architecture alleviates the problem of overfitting. During training, when the dropout layer was not used, after 10 epochs, the training accuracy reached 99% while the validation accuracy was only 85%. But training the modified model with a dropout layer overcame this problem. The graph displayed in Figure 5.16 (a) illustrates that there is a vast difference in training and validation loss when the dropout layer is not present (*i.e.*, overfitting occurs). But the graph shown in Figure 5.16 (b) demonstrates that there is no overfitting in the model.

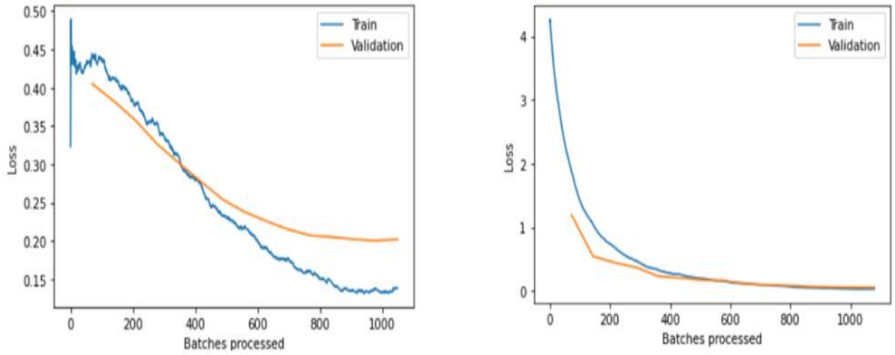


Figure 5.16 Training and Validation Loss: (a) Without Dropout Layer (b) With Dropout Layer

c) Effect of learning rate on recognition accuracy: The use of a fixed learning rate may prevent improvement in validation accuracy. The difference in training and validation losses shown in Figure 5.17 illustrates the effect of fixed learning rates and cyclical learning rates on the performance of the model. Figure 5.17 (b) shows the parallel improvement of both the training and validation accuracy. The learning rate finder curve generated during experiments shown in Figure 4.9 has been used to find out the optimized learning rate. The use of a cyclical learning rate helps the model achieve better performance.

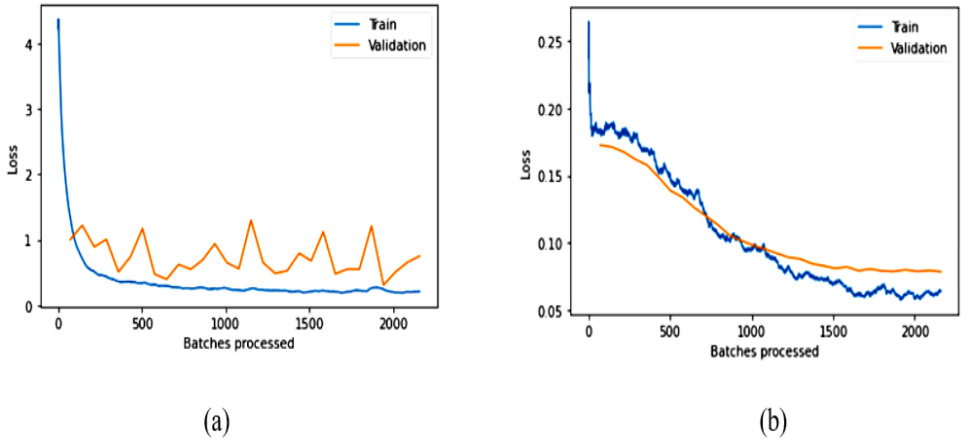


Figure 5.17 Training and Validation Loss using: (a) Fixed Learning Rate (b) Cyclical Learning Rate

d) Effect of the concatenation of GAP and GMP on recognition accuracy: It is observed that sometimes the maximum value of the feature map received from the previous layers gives better results, and sometimes the average value is good. Therefore, the present work concatenated the global average pooling and global max pooling to get the optimized value for better performance. The recognition accuracy achieved after the concatenation is higher than the recognition accuracy with the use of only the global average pooling layer, as shown in Table 5.10.

e) Effect of optimizer on recognition accuracy: The model has been evaluated on two optimizers, such as Adam and Stochastic Gradient Descent (SGD), to highlight the effect of selecting the optimizer for the network. The modified models have achieved higher accuracy with the Adam optimizer in comparison to SGD. Therefore, we used the Adam optimizer in the proposed modified models.

f) Effect of activation function on recognition accuracy: The usage of Leaky ReLU after the BN layer yields better results in the present work, so we used the Leaky ReLU activation function because it mitigates the problem of dying ReLU.

g) Effect of the addition of a fully connected layer on recognition accuracy: The motivation for the addition of a fully connected layer is discussed in Table 5.10, and the addition of one fully connected layer in the network enhances the recognition accuracy of the pre-trained models.

5.2.3 Alert Generation

In this module, the location of the GPS-enabled CCTV camera has been identified after the successful recognition of any suspect, as shown in Figure 5.18. In this work, the camera of our system has been used for experimental purposes. Once the criminal has been identified and recognized using the module given in Section 5.2.2, the nearest police station from the current location of the criminal is searched using the Haversine formula given in equation (5.7), and automatically, it sends an alert via mail and message to the registered email ID and contact number

of the police official stored in the database given in Figure 5.3 (b). The Haversine formula has been used to calculate the shortest distance between the two locations on the sphere using their latitudes and longitudes. The information, photograph, and current location of the criminal are sent to the registered email ID, and the location is also sent to the registered mobile number, as shown in Figure 5.19.

$$H = 2r\sin^{-1}(\sqrt{sin^2\left(\frac{a2-a1}{2}\right) + \cos(a1)\cos(a2)\,sin^2(\frac{\varepsilon 2-\varepsilon 1}{2})} \qquad (5.7)$$

Where r is taken as the earth's radius (6371 km), the distance between the two location points on the earth is H; a1, a2 are considered the latitudes of the two points, and ε1, ε 2 are taken as the longitudes of the two points on the earth.

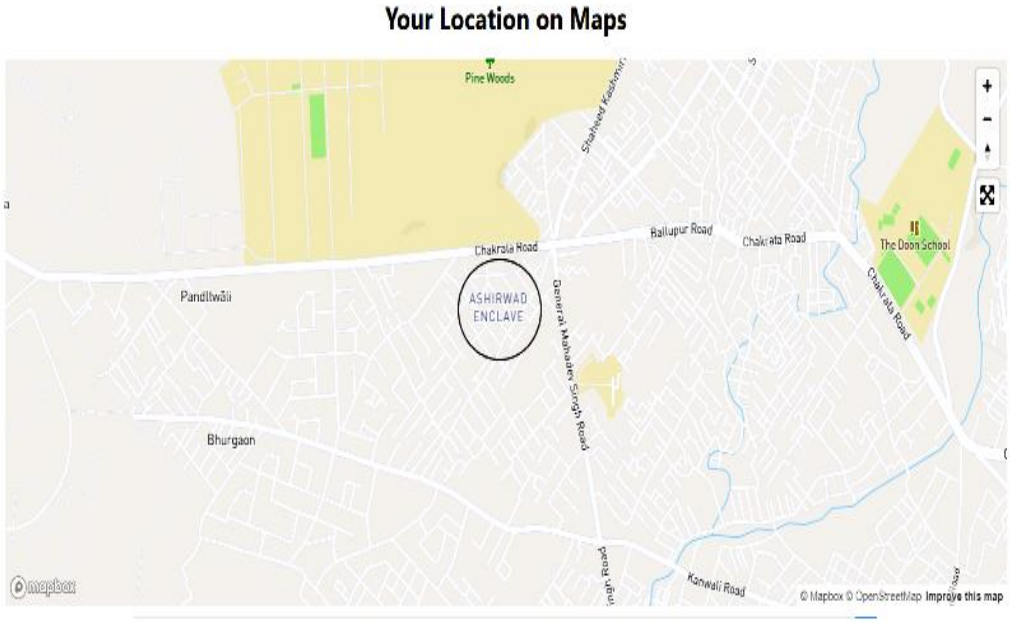
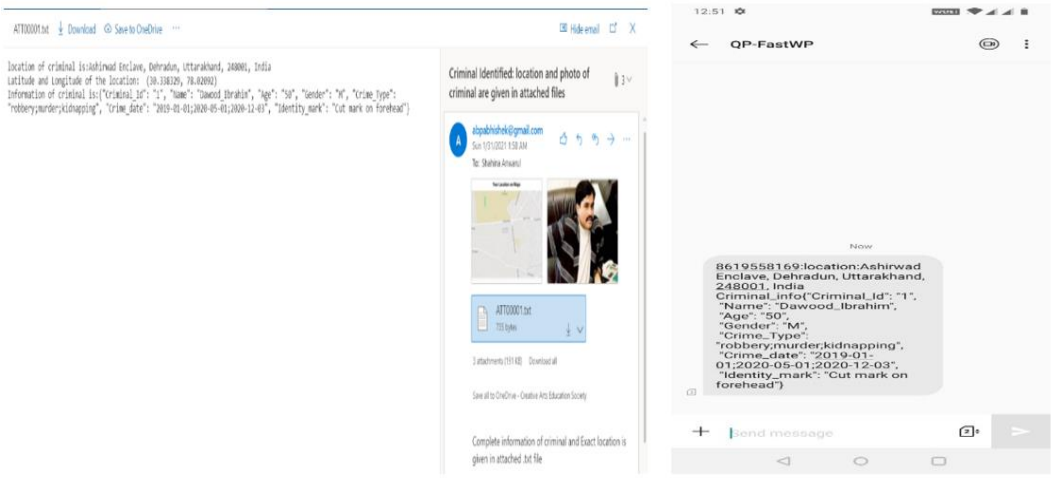


Figure 5.18 Current Location of Criminal



(a) (b)

Figure 5.19 Alert Generation via: (a) Mail and (b) Message

5.2.4 Prediction of Crime Prone Areas

All the identified locations (*i.e.*, latitude and longitude) of criminals have been saved in a file given in Figure 5.20 in parallel once the criminal has identified in any location to cluster the crime-prone regions. On a weekly basis, the system generates the crime regions' clusters and shows those regions on Google Maps. This module of the proposed criminal recognition system helps police officials analyze crime-prone areas. K-means clustering has been utilized to identify the clusters of crime-prone regions (areas where most of the criminals are identified), as highlighted in red in Figure 5.21.

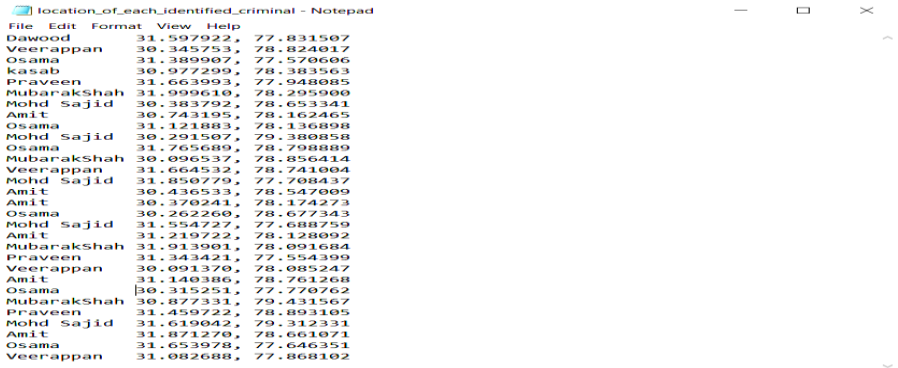


Figure 5.20 Location of Identified Criminals

Figure 5.21 Clusters of the Crime Prone Regions

5.2.5 Time Complexity Analysis

In the realm of research, it carries substantial importance to encompass both an algorithm's actual execution time and its time complexity. The quantifiable execution time offers a clear indication of the algorithm's performance in real-world settings. This can be determined by measuring the duration of the completion of a particular activity or the processing of each individual data element. In contrast, an algorithm's time complexity shows how much processing time is required in relation to the size of the input. This dimension imparts an abstract vantage point on the algorithm's efficiency. Time complexity is frequently conveyed through the big O notation, which establishes an upper limit on the algorithm's runtime within the worst-case scenario. This notation aids in comprehending how the algorithm's efficiency aligns with other methods. Thus, the integration of both the actual execution time and the time complexity in research engenders a holistic comprehension of the algorithm's real-world performance and its efficiency in relation to alternative approaches.

In the proposed system, the time complexity is contingent upon the participant algorithms, primarily focusing on the maximum execution time denoted as T. Here, T represents the maximum value among $T_{SSD,}$ $T_{HE\text{-}CNN,}$ and T_A. In the first phase, SSD is implemented for the detection of faces; let's say it will take time of $O(T_{SSD})$. The execution time of the second phase, *i.e.*, the recognition stage, depends on the execution time of HE-CNN, say $O(T_{HE\text{-}CNN})$. Last, the third stage of alert generation takes time of $O(T_A)$. So, the total time complexity of the system is represented by equation (5.8).

$$O(T) = O(T_{SSD}) + O(T_{HE\text{-}CNN}) + O(T_A) \quad (5.8)$$

Here, $O(T_{SSD})$ is O(n), where n is the number of generated bounding boxes. In the proposed HE-CNN model, the assembling of three modified pre-trained models are used with time complexities of $O(N_1)$, $O(N_2)$, and $O(N_3)$, where $N_{1,}$ $N_{2,}$ N_3 is the number of operations required to process a single input image through the network. After that, a weighted sum operation is used for the final prediction. The time complexity of the weighted sum operation in ensemble learning depends on the number of models in the ensemble and the size of the input data. Let's assume we have "x" models in the ensemble, and each model takes an input of size "m". In the weighted sum operation, we multiply each model's prediction by its corresponding weight and sum them up. In the worst case, the time complexity of the weighted sum operation can be expressed as O(x * m), where "x" is the number of models and "m" is the size of the input data. Therefore, $T_{HE\text{-}CNN}$ is the maximum of $N_{1,}$ $N_{2,}$ N_3, and (x*m). The time taken by the alert generation phase *i.e.*, $O(T_A)$ depends on the number of deployed CCTV cameras and the number of registered police stations. The Haversine formula's time complexity is denoted as O(1). Consider a scenario where C cameras are deployed within a city and P police stations are officially registered. Consequently, the overall time taken to compute the minimum distance using the Haversine formula is represented by O(P * C * 1).

The time duration for a face recognition algorithm's execution is influenced by various factors, including the input image's dimensions, algorithm complexity, system processing capabilities, and implementation efficiency. Generally, the execution time for face recognition algorithms tends to elongate with larger input image sizes. This outcome arises from the increased computational requirements for face detection and recognition operations in larger images. Moreover, the complexity of the algorithm and the processing power of the underlying system play pivotal roles in determining the execution time. A more intricate algorithm or a system with limited processing capabilities could lead to extended execution times. Furthermore, the efficiency of the implementation itself exerts a notable influence on the execution time of the face recognition algorithm. Implementations that are well optimized can minimize superfluous computations and enhance overall performance. Providing an exact execution time in minutes and seconds for a face recognition algorithm is challenging, as it varies based on the aforementioned factors. The best method for figuring out how long a certain implementation will take to execute is to evaluate its performance using a representative dataset. This empirical assessment delivers a more precise estimate of the algorithm's execution time in real-world scenarios. These experiments were conducted on a system equipped with Windows 10, 16GB of RAM, and an NVIDIA GTX 1650 Ti with a 4GB GPU. The conducted experiments encompass standard datasets as well as self-curated dataset, collectively shedding light on the comprehensive execution performance of the proposed system. On average, model training necessitated around 1 hour and 15 minutes across various parametric configurations for the considered datasets. It required approximately 45 seconds to test a single image with a resolution of 1024 by 1024 pixels.

5.3 Summary of the Chapter

In this chapter, an outline of the experimental setup and design for the automated face recognition system is presented. Utilizing the ensemble learning-

based HE-CNN model, we extracted results from diverse datasets, including LFW, CPLFW, GT face, YTF, and a self-created dataset. The assessment of the system's efficacy involves the utilization of evaluation parameters such as precision, recall, accuracy, error rate, and ROC curve. These metrics enable a comprehensive comparative analysis between the present work and the current state-of-the-art methods. The findings of the present research unequivocally underscore the superior performance of our proposed technique in contrast to prevailing deep learning methods for face recognition. Furthermore, a meticulous ablation study is undertaken on the presented approach, unraveling the effects of the modifications made to the pre-trained models integrated into the HE-CNN model. This analysis enhances the understanding of the specific enhancements contributed by these modifications. Acknowledging the significance of computational efficiency, we subject the present approach to an evaluation of its time complexity. This assessment encompasses both the real-world execution time and the theoretical time complexity of the algorithm, providing insights into its computational efficiency. The subsequent chapter will encapsulate the conclusions derived from this research endeavor and delineate potential trajectories for future exploration in this evolving domain.

CHAPTER-6

CONCLUSIONS AND FUTURE DIRECTIONS

Face recognition is a challenging task in video surveillance due to the presence of various unconstrained factors such as pose variation, occlusion, illumination, and low resolution. There is a need for continuous monitoring of CCTV footage to identify the individual in recordings of existing face recognition systems. The proposed recognition system helps concerned officials monitor the surveillance area without human intervention. It automatically alerts police officials when there is an identification of criminals in a specified area. It also helps to prevent crimes by providing clusters of crime-prone areas. Due to this, the police officials will become attentive before the crime happens in those areas. The extensive developments in face recognition in recent years have given immense scope to criminal identification and other applications. The emergence of deep learning has made recognition systems accurate, but it requires a large dataset for training the machine. The non-availability of a considerable number of images of criminals limits the accuracy of current systems. Therefore, the advocated solution given in the research work takes advantage of transfer learning and extracts features from the modified models trained on the ImageNet dataset that can be executed with less data. The designed HE-CNN model using modified pre-trained models is successfully validated on standard datasets and a self-curated dataset of mugshots. In this study, it became evident that the transfer learning approach surpasses conventional methods in terms of performance. The development of the proposed hybrid architecture, known as the HE-CNN model, for face recognition is based on extensive experimentation. Given the challenges associated with collecting a significant volume of face images due to privacy considerations, the present work offers an optimal solution through the implementation of deep ensemble transfer learning. The primary driving force behind this research was to design and develop an automated face recognition system based on ensemble deep learning methods with

superior accuracy and minimal complexity. This chapter highlights the major contributions aligned with the objectives of the research and discusses possible future research directions.

6.1 Summary of the Thesis and Objective Attenuation

In this thesis, we present an ensemble learning-based model to be applied to an automated face recognition system. We provided one self-curated dataset of mugshots to demonstrate the real-time application of the proposed research. For other researchers working in this field, we have made the dataset available in the public domain [235]. The main research objectives mentioned in Section 1.3 of Chapter 1 have been addressed in this thesis in the following order:

The initial stage of face recognition systems involves face detection. In this context, we assessed four distinct algorithms: the Haar Feature-based cascade classifier (Viola Jones) and the Local Binary Pattern (LBP) Feature-based cascade classifier, as well as the MTCNN and SSD, which are both deep learning-based methods. These algorithms were evaluated for their effectiveness in detecting faces within video frames. During the evaluation, key metrics such as True Positive Rate (TPR), False Positive Rate (FPR), and False Negative Rate (FNR) were considered. The algorithm that exhibited the optimal balance of attributes, including fast execution time, higher TPR, and lower FPR and FNR, was selected to serve in the face detection phase within the system. Thus, the first sub-objective of this research work has been achieved.

A hybrid ensemble CNN model (HE-CNN) is proposed for the recognition of an individual in a video frame. We modified the backbone architecture's last-level classification layer by replacing it with a linear combination of the concatenation of global average pooling and global max pooling. In our investigations, the maximum and average activations from the previous convolution are preserved, offering the model knowledge of both approaches and enhancing performance.

Furthermore, our model leveraged pre-trained models from the ImageNet dataset to extract features. This approach effectively mitigates the computational overhead and extensive data prerequisites associated with training the model entirely from scratch. The two-phase training process and the use of hyperparameter tuning help the proposed model achieve SOTA-competent accuracy on benchmark datasets. The evaluation of the proposed model is done using different evaluation parameters such as accuracy, recall, precision, error rate, and ROC curve. The proposed model achieved an accuracy of 99.35% in LFW, 91.58% in CPLFW, 99.63% in GT face, 99.21% in YTF, and 95% in a self-curated dataset. Therefore, the second sub-objective of this research work has been successfully attained.

This research work introduced a system for alert generation designed to streamline the process of individual recognition, thereby reducing the need for extensive human involvement. This system initiates the transmission of alert emails and messages to the nearest registered police station. The identification of the nearest police station is achieved utilizing the Haversine formula, a method integrated into the implemented system. To facilitate further insights, the system also records the location coordinates of the recognized individual. This information is stored in a dedicated file, contributing to the establishment of clusters in areas where a higher frequency of target individuals has been identified. This approach has the potential to heighten the vigilance of law enforcement personnel in regions characterized by an elevated likelihood of criminal activity. Thus, the third sub-objective of this research has been achieved.

6.2 Research Future Directions

The proposed HE-CNN model for face recognition provides better recognition accuracy than the baseline models due to the effectiveness of ensemble learning-based models. However, the research presented here has a wider scope, with several extensions addressing a variety of challenges that require future attention, as is the case with many other academic articles in the same field. In the part that follows,

we go through some of these issues and suggest upcoming directions that, in our opinion, will have a significant influence.

In this research, we utilized a single laptop camera operating at a refresh rate of 60.02 Hz for individual recognition, and benchmark datasets used in the existing research were generated using one or more cameras. The diverse cameras with varying refresh rates can be used to establish the optimal refresh rate. This threshold identification aims to minimize data loss during the recognition process.

The expansion of the self-curated dataset can involve procuring additional image samples for each class through free sources on the Internet. Alternatively, employing various data augmentation techniques, like elastic deformation to replicate distortions and stretches or introducing Gaussian noise, can aid the model in acquiring the ability to discern objects within noisy environments.

It is essential to acknowledge a limitation of our system: it can identify individuals based on images presented in front of the camera, rendering it susceptible to spoofing. Therefore, an extension of our work could involve identifying spoofed faces to enhance security.

To expand this study, the inclusion of other video datasets in the experimental evaluation could provide comprehensive insights.

Additionally, opportunities for collaboration with government and law enforcement entities could lead to large-scale implementation of the research.

In existing research, face recognition models are typically assessed through diverse metrics including accuracy, precision, recall, F1-score, and ROC curve. However, supplementing these evaluations with statistical analysis can fortify the research findings, facilitate hypothesis testing, and unveil hidden insights within the data.

Furthermore, it is worth considering the incorporation of alternative biometric modalities or soft biometric characteristics as supplementary sources of data. This

approach can contribute to the development of face recognition systems that are both more dependable and precise, aligning better with the demands of real-world video surveillance applications.

REFERENCES

[1] Xiao, K., Tian, Y., Lu, Y., Lai, Y., & Wang, X. (2022). Quality assessment-based iris and face fusion recognition with dynamic weight. *The Visual Computer*, 1-13.

[2] Min-Allah, N., Jan, F., & Alrashed, S. (2021). Pupil detection schemes in human eye: a review. *Multimedia Systems*, 27(4), 753-777.

[3] Kortli, Y., Jridi, M., Al Falou, A., & Atri, M. (2020). Face recognition systems: A survey. *Sensors*, 20(2), 342.

[4] Sharif, M., Naz, F., Yasmin, M., Shahid, M. A., & Rehman, A. (2017). Face Recognition: A Survey. *Journal of Engineering Science & Technology Review*, 10(2).

[5] Patel, R., Rathod, N., & Shah, A. (2012). Comparative analysis of face recognition approaches: a survey. *International Journal of Computer Applications*, 57(17).

[6] Lal, M., Kumar, K., Arain, R. H., Maitlo, A., Ruk, S. A., & Shaikh, H. (2018). Study of face recognition techniques: a survey. *International Journal of Advanced Computer Science and Applications*, 9(6).

[7] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4), 399-458.

[8] Tan, X., Chen, S., Zhou, Z. H., & Zhang, F. (2006). Face recognition from a single image per person: A survey. *Pattern recognition*, 39(9), 1725-1745.

[9] Pagano, C., Granger, E., Sabourin, R., Marcialis, G. L., & Roli, F. (2014). Adaptive ensembles for face recognition in changing video surveillance environments. *Information sciences*, 286, 75-101.

[10] Khan, A., Rehman, S., Waleed, M., Khan, A., Khan, U., Kamal, T., ... & Marwat, S. N. K. (2018). Forensic video analysis: passive tracking system for automated Person of Interest (POI) localization. *IEEE Access*, 6, 43392-43403.

[11] Dhamija, J., Choudhury, T., Kumar, P., & Rathore, Y. S. (2017, October). An Advancement towards Efficient Face Recognition Using Live Video Feed:" For the Future". In *2017 3rd International conference on computational intelligence and networks (CINE)* (pp. 53-56). IEEE.

[12] Kavitha, J., & Mirnalinee, T. T. (2016). Automatic frontal face reconstruction approach for pose invariant face recognition. *Procedia Computer Science*, 87, 300-305.

[13] Liu, F., Zhao, Q., Liu, X., & Zeng, D. (2018). Joint face alignment and 3D face reconstruction with application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 42(3), 664-678.

[14] Mudunuri, S. P., & Biswas, S. (2015). Low resolution face recognition across variations in pose and illumination. *IEEE transactions on pattern analysis and machine intelligence*, 38(5), 1034-1040.

[15] Huang, Y. H., & Chen, H. H. (2020, October). Face recognition under low illumination via deep feature reconstruction network. In *2020 IEEE International Conference on Image Processing (ICIP)* (pp. 2161-2165). IEEE.

[16] Zeng, D., Veldhuis, R., & Spreeuwers, L. (2021). A survey of face recognition techniques under occlusion. *IET biometrics*, 10(6), 581-606.

[17] Joshi, D., Anwarul, S., & Mishra, V. (2020). Deep leaning using keras. In *Machine Learning and Deep Learning in Real-Time Applications* (pp. 33-60). IGI Global.

[18] Anwarul, S., & Joshi, D. (2020). Deep learning with tensorflow. In *Machine Learning and Deep Learning in Real-Time Applications* (pp. 96-120). IGI Global.

[19] Masi, I., Wu, Y., Hassner, T., & Natarajan, P. (2018, October). Deep face recognition: A survey. In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)* (pp. 471-478). IEEE.

[20] Shailendra, R., Jayapalan, A., Velayutham, S., Baladhandapani, A., Srivastava, A., Kumar Gupta, S., & Kumar, M. (2022). An IoT and machine learning based intelligent system for the classification of therapeutic plants. *Neural Processing Letters*, 54(5), 4465-4493.

[21] Madhu, G., Govardhan, A., Ravi, V., Kautish, S., Srinivas, B. S., Chaudhary, T., & Kumar, M. (2022). DSCN-net: a deep Siamese capsule neural network model for automatic diagnosis of malaria parasites detection. *Multimedia Tools and Applications*, 81(23), 34105-34127.

[22] Stark, L., Stanhaus, A., & Anthony, D. L. (2020). “i don't want someone to watch me while i'm working”: Gendered views of facial recognition technology in workplace surveillance. *Journal of the Association for Information Science and Technology*, 71(9), 1074-1088.

[23] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27* (pp. 270-279). Springer International Publishing.

[24] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345-1359.

[25] Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115, 105151.

[26] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: Aunified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).

[27] Sun, Y., Wang, X., & Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1891-1898).

[28] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018, May). Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)* (pp. 67-74). IEEE.

[29] Starlink (2021). Facial Recognition Technology: Functionality, Applications, & Significance in Today's World. [Online] Available: *https://www.starlinkindia.com/blog/facial-recognition-technology-*

functionality-applications-significance-in-todays-world/. [Accessed 1 July 2023].

[30] Ding, C., & Tao, D. (2017). Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 1002-1014.

[31] Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1701-1708).

[32] Abdelmaksoud, M., Nabil, E., Farag, I., & Hameed, H. A. (2020). A novel neural network method for face recognition with a single sample per person. *IEEE Access*, 8, 102212-102221.

[33] Xu, J. (2021). A deep learning approach to building an intelligent video surveillance system. *Multimedia Tools and Applications*, 80(4), 5495-5515.

[34] Bagchi, T., Mahapatra, A., Yadav, D., Mishra, D., Pandey, A., Chandrasekhar, P., & Kumar, A. (2022). Intelligent security system based on face recognition and IoT. *Materials Today: Proceedings*, 62, 2133-2137.

[35] Said, Y., Barr, M., & Ahmed, H. E. (2020). Design of a face recognition system based on convolutional neural network (CNN). *Engineering, Technology & Applied Science Research*, 10(3), 5608-5612.

[36] Yang, H., & Han, X. (2020). Face recognition attendance system based on real-time video processing. *IEEE Access*, 8, 159143-159150.

[37] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV*

2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

[38] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10), 1499-1503.

[39] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57, 137-154.

[40] Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12), 2037-2041.

[41] Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008, October). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*.

[42] Zheng, T., & Deng, W. (2018). Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications*, Tech. Rep, 5(7).

[43] Georgia Tech Face dataset. [Online] Available: *http://www.anefian.com/research/face_reco.htm*

[44] Wolf, L., Hassner, T., & Maoz, I. (2011, June). Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011* (pp. 529-534). IEEE.

[45] Gao, Y., & Lee, H. J. (2015). Cross-pose face recognition based on multiple virtual views and alignment error. *Pattern Recognition Letters*, 65, 170-176.

[46] Abdullah, N. A., Saidi, M. J., Rahman, N. H. A., Wen, C. C., & Hamid, I. R. A. (2017, October). Face recognition for criminal identification: An implementation of principal component analysis for face recognition. In *AIP conference proceedings* (Vol. 1891, No. 1, p. 020002). AIP Publishing LLC.

[47] Baykara, M., & Daş, R. (2013, November). Real time face recognition and tracking system. In *2013 International Conference on Electronics, Computer and Computation (ICECCO)* (pp. 159-163). IEEE.

[48] Chawla, D., & Trivedi, M. C. (2018). A comparative study on face detection techniques for security surveillance. In *Advances in Computer and Computational Sciences: Proceedings of ICCCCS 2016*, Volume 2 (pp. 531-541). Springer Singapore.

[49] Kakkar, P., & Sharma, V. (2018). Criminal identification system using face detection and recognition. *International Journal of Advanced Research in Computer and Communication Engineering*, 7(3), 238-243.

[50] Sable, A. H., Talbar, S. N., & Dhirbasi, H. A. (2019). Recognition of plastic surgery faces and the surgery types: An approach with entropy based scale invariant features. *Journal of King Saud University-Computer and Information Sciences*, 31(4), 554-560.

[51] Banerjee, S., Brogan, J., Krizaj, J., Bharati, A., Webster, B. R., Struc, V., ... & Scheirer, W. J. (2018, March). To frontalize or not to frontalize: Do we really need elaborate pre-processing to improve face recognition?. In *2018*

IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 20-29). IEEE.

[52] Huang, Z., Shan, S., Wang, R., Zhang, H., Lao, S., Kuerban, A., & Chen, X. (2015). A benchmark and comparative study of video-based face recognition on cox face database. *IEEE Transactions on Image Processing*, 24(12), 5967-5981.

[53] Fathima, A. A., Ajitha, S., Vaidehi, V., Hemalatha, M., Karthigaiveni, R., & Kumar, R. (2015, November). Hybrid approach for face recognition combining gabor wavelet and linear discriminant analysis. In *2015 IEEE international conference on computer graphics, vision and information security (CGVIS)* (pp. 220-225). IEEE.

[54] Lei, Z., Wang, C., Wang, Q., & Huang, Y. (2009, March). Real-time face detection and recognition for video surveillance applications. In *2009 WRI World Congress on Computer Science and Information Engineering* (Vol. 5, pp. 168-172). IEEE.

[55] Borysiuk, Z., Konieczny, M., Krecisz, K., & Pakosz, P. (2018). Application of sEMG and Posturography as Tools in the Analysis of Biosignals of Aging Process of Subjects in the Post-production Age. In *Biomedical Engineering and Neuroscience: Proceedings of the 3rd International Scientific Conference on Brain-Computer Interfaces, BCI 2018, March 13-14, Opole, Poland* (pp. 23-29). Springer International Publishing.

[56] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5), 1-36.

[57] Dormehl, L. (2019). What is an artificial neural network? Here's everything you need to know. [Online]. Available: *https://www.digitaltrends.com/cool-tech/what-is-anartificial-neural-network*. [Accessed 2 July 2023].

[58] Dargan, S., Kumar, M., Ayyagari, M. R., & Kumar, G. (2020). A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27, 1071-1092.

[59] hajan, P., Abrol, P., & Lehana, P. K. (2020). Scene based classification of aerial images using convolution neural networks. *Journal of Scientific & Industrial Research*, 79(12), 1087-1094.

[60] Prabhu (2018). Understanding of Convolutional Neural Network (CNN)— Deep Learning. [Online]. Available: *https://medium.com/@RaghavPrabhu/understanding-ofconvolutional-neural-network-cnn-deep-learning-2099760835f148*. [Accessed 3 July 2023].

[61] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.

[62] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

[63] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

[64] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211-252.

[65] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

[66] Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *Computing Research Repository (CoRR)*, 1312.4400.

[67] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)* (pp. 1-14).

[68] He, K., & Sun, J. (2015). Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5353-5360).

[69] Quinn, J., McEachen, J., Fullan, M., Gardner, M., & Drummy, M. (2019). *Dive into deep learning: Tools for engagement*. Corwin Press.

[70] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[71] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[72] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

[73] Tsang, S. (2018). Review: DenseNet-Dense Convolutional Network (Image Classification). [Online]. Available: *https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803*. [Accessed 5 July 2023].

[74] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., ... & Ferrari, V. (2020). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7), 1956-1981.

[75] Ouyang, W., Wang, X., Zeng, X., Qiu, S., Luo, P., Tian, Y., ... & Tang, X. (2015). Deepid-net: Deformable deep convolutional neural networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2403-2412).

[76] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 764-773).

[77] Wang, L., Ouyang, W., Wang, X., & Lu, H. (2015). Visual tracking with fully convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 3119-3127).

[78] T. N., & Sugimoto, A. (2017, September). Deeply Supervised 3D Recurrent FCN for Salient Object Detection in Videos. In *BMVC* (Vol. 1, p. 3).

[79] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014, January). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (pp. 647-655). PMLR.

[80] Azizpour, H., Sharif Razavian, A., Sullivan, J., Maki, A., & Carlsson, S. (2015). From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 36-45).

[81] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

[82] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 834-848.

[83] Pohlen, T., Hermans, A., Mathias, M., & Leibe, B. (2017). Full-resolution residual networks for semantic segmentation in street scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4151-4160).

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

[85] Denton, E. L., Zaremba, W., Bruna, J., LeCun, Y., & Fergus, R. (2014). Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27.

[86] Girshick, R. (2015). Fast R-CNN. Proceedings of the *IEEE International Conference on Computer Vision (ICCV)*.

[87] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

[88] Adriana, R., Nicolas, B., Ebrahimi, K. S., Antoine, C., Carlo, G., & Yoshua, B. (2015). Fitnets: Hints for thin deep nets. *Proc. ICLR*, 2.

[89] Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, *28*.

[90] Luo, J. H., Wu, J., & Lin, W. (2017). Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision* (pp. 5058-5066).

[91] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

[92] Yang, Z., Moczulski, M., Denil, M., De Freitas, N., Smola, A., Song, L., & Wang, Z. (2015). Deep fried convnets. In *Proceedings of the IEEE international conference on computer vision* (pp. 1476-1483).

[93] Chellappa, R., Chen, J. C., Ranjan, R., Sankaranarayanan, S., Kumar, A., Patel, V. M., Castillo, C. D. (2016, January). Towards the design of an end-

to-end automated system for image and video-based recognition. In *2016 Information Theory and Applications Workshop (ITA)* (pp. 1-7). IEEE.

[94] Huang, G. B., Lee, H., & Learned-Miller, E. (2012, June). Learning hierarchical representations for face verification with convolutional deep belief networks. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2518-2525). IEEE.

[95] Sun, Y., Chen, Y., Wang, X., & Tang, X. (2014). Deep learning face representation by joint identification-verification. *Advances in neural information processing systems*, 27.

[96] Sun, Y., Wang, X., & Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2892-2900).

[97] Sun, Y., Wang, X., & Tang, X. (2013). Hybrid deep learning for face verification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1489-1496).

[98] Parkhi, O. M., Vedaldi, A. & Zisserman, A. (2015). Deep face recognition. *BMVC*.

[99] Yue, G., & Lu, L. (2018, August). Face recognition based on histogram equalization and convolution neural network. In *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 1, pp. 336-339). IEEE.

[100] Parchami, M., Bashbaghi, S., & Granger, E. (2017, May). Video-based face recognition using ensemble of haar-like deep convolutional neural networks.

In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 4625-4632). IEEE.

[101] Parchami, M., Bashbaghi, S., & Granger, E. (2017, August). Cnns with crosscorrelation matching for face recognition in video surveillance using a single training sample per person. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.

[102] Gao, S., Zhang, Y., Jia, K., Lu, J., & Zhang, Y. (2015). Single sample face recognition via learning deep supervised autoencoders. *IEEE transactions on information forensics and security*, 10(10), 2108-2118.

[103] Parchami, M., Bashbaghi, S., Granger, E., & Sayed, S. (2017, August). Using deep autoencoders to learn robust domain-invariant representations for still-to-video face recognition. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.

[104] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12).

[105] Kan, M., Shan, S., Chang, H., & Chen, X. (2014). Stacked progressive autoencoders (spae) for face recognition across poses. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1883-1890).

[106] Le, Q.V. (2013, May). Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8595-8598). IEEE.

[107] Zhu, Z., Luo, P., Wang, X., & Tang, X. (2014). Recover canonical-view faces in the wild with deep neural networks. *arXiv preprint arXiv:1404.3543*.

[108] Zhu, Z., Luo, P., Wang, X., & Tang, X. (2014). Multi-view perceptron: a deep model for learning face identity and view representations. *Advances in neural information processing systems*, 27.

[109] Ghodrati, A., Jia, X., Pedersoli, M., & Tuytelaars, T. (2015). Towards automatic image editing: Learning to see another you. *arXiv preprint arXiv:1511.08446*.

[110] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., ... & Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2758-2766).

[111] Huang, R., Zhang, S., Li, T., & He, R. (2017). Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *proceedings of the IEEE international conference on computer vision* (pp. 2439-2448).

[112] Tran, L., Yin, X., & Liu, X. (2017). Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1415-1424).

[113] Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14* (pp. 499-515). Springer International Publishing.

[114] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. *Information*, 11(2), 125.

[115] Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep learning techniques: an overview. *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020*, 599-608.

[116] Panigrahi, A., Chen, Y., & Kuo, C. C. J. (2018). Analysis on gradient propagation in batch normalized residual networks. *arXiv preprint arXiv:1812.00342.*

[117] Lorraine, J., & Duvenaud, D. (2018). Stochastic hyperparameter optimization through hypernetworks. *arXiv preprint arXiv:1802.09419.*

[118] Vieira, A., & Ribeiro, B. (2018). Introduction to deep learning business applications for developers. *Berkeley, CA, USA: Apress*.

[119] Takahashi, T. (2018). U.S. Patent No. 10,013,644. *Washington, DC: U.S. Patent and Trademark Office*.

[120] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.

[121] Achille, A., & Soatto, S. (2018). Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12), 2897-2905.

[122] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). pmlr.

[123] Polikar, R. (2012). Ensemble learning. *Ensemble machine learning: Methods and applications*, 1-34.

[124] Sharma, P. (2019). Top 5 deep learning frameworks, their applications, and comparisons. [Online]. Available: *https://www.analyticsvidhya.com/blog/2019/03/deep-learning-frameworks-comparison/*. [Accessed 8 July 2023].

[125] Howard, J., & Gugger, S. (2020). Fastai: A layered API for deep learning. *Information*, 11(2), 108.

[126] Baheti, P. (2021). A Newbie-Friendly Guide to Transfer Learning. [Online] Available: *https://www.v7labs.com/blog/transfer-learning-guide#h1*. [Accessed 10 July 2023].

[127] Niu, S., Liu, Y.,Wang, J., &Song, H. (2020). A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, 1(2), 151-166.

[128] Hu, J. (2017). Discriminative transfer learning with sparsity regularization for single-sample face recognition. *Image and vision computing*, 60, 48-57.

[129] Alhanaee, K., Alhammadi, M., Almenhali, N., & Shatnawi, M. (2021). Face recognition smart attendance system using deep transfer learning. *Procedia Computer Science*, 192, 4093-4102.

[130] Antipov, G., Berrani, S. A., & Dugelay, J. L. (2016). Minimalistic CNN-based ensemble model for gender prediction from face images. *Pattern recognition letters*, 70, 59-65.

[131] Minetto, R., Segundo, M. P., & Sarkar, S. (2019). Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9), 6530-6541.

[132] Ding, C., & Tao, D. (2017). Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 1002-1014.

[133] Kumar, A., Kim, J., Lyndon, D., Fulham, M., & Feng, D. (2016). An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE journal of biomedical and health informatics*, 21(1), 31-40.

[134] Wang, H. Z., Li, G. Q., Wang, G. B., Peng, J. C., Jiang, H., & Liu, Y. T. (2017). Deep learning based ensemble approach for probabilistic wind power forecasting. *Applied energy*, 188, 56-70.

[135] Töscher, A., Jahrer, M., & Bell, R. M. (2009). The bigchaos solution to the netflix grand prize. *Netflix prize documentation*, 1-52.

[136] Tang, J., Su, Q., Su, B., Fong, S., Cao,W., & Gong, X. (2020). Parallel ensemble learning of convolutional neural networks and local binary

patterns for face recognition. *Computer Methods and Programs in Biomedicine*, 197, 105622.

[137] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4), 463-484.

[138] Granger, E., Khreich, W., Sabourin, R., & Gorodnichy, D. O. (2012). Fusion of biometric systems using boolean combination: an application to iris-based authentication. *International journal of biometrics*, 4(3), 291-315.

[139] Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2), 181.

[140] Zhu, Y., Li, Y., Mu, G., Shan, S., & Guo, G. (2016). Still-to-video face matching using multiple geodesic flows. *IEEE Transactions on Information Forensics and Security*, 11(12), 2866-2875.

[141] Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2), 241-259.

[142] Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super learner. *statistical applications in genetics and molecular biology*, 6(1).

[143] Canziani, A., Paszke, A., & Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.

[144] Khade, B. S., Gaikwad, H. M., Aher, A. S., &Patil, K. 3. (2016). Face recognition techniques: a survey. *International Journal of Computer Science and Mobile Computing*, 5(11), 65-72.

[145] Fu, T. C., Chiu, W. C., & Wang, Y. C. F. (2017, September). Learning guided convolutional neural networks for cross-resolution face recognition. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1-5). IEEE.

[146] Zhang, W., Shan, S., Chen, X., & Gao, W. (2007). Local Gabor binary patterns based on Kullback–Leibler divergence for partially occluded face recognition. *IEEE signal processing letters*, 14(11), 875-878.

[147] Zou, W.W., & Yuen, P. C. (2011). Very low resolution face recognition problem. *IEEE Transactions on image processing*, 21(1), 327-340.

[148] Tin, H. H. K. (2011). Removal of noise by median filtering in image processing. *6th Parallel Soft Comput. (PSC 2011)*, 1-3.

[149] Luo, Z., Hu, J., Deng, W., & Shen, H. (2018, May). Deep unsupervised domain adaptation for face recognition. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)* (pp. 453-457). IEEE.

[150] Wang, P., Lin, W. H., Chao, K. M., & Lo, C. C. (2017, November). A face recognition approach using deep reinforcement learning approach for user authentication. In *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)* (pp. 183-188). IEEE.

[151] S. Z., Chu, R., Liao, S., & Zhang, L. (2007). Illumination invariant face recognition using near-infrared images. *IEEE Transactions on pattern*

analysis and machine intelligence, 29(4), 627-639.

[152] Ding, C., Xu, C., & Tao, D. (2015). Multi-task pose-invariant face recognition. *IEEE Transactions on image Processing*, 24(3), 980-993.

[153] Singh, R., Vatsa, M., Ross, A., & Noore, A. (2007). A mosaicing scheme for pose invariant face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5), 1212-1225.

[154] Zhou, H., & Lam, K. M. (2018). Age-invariant face recognition based on identity inference from appearance age. *Pattern recognition*, 76, 191-202.

[155] Li, Z., Park, U., & Jain, A. K. (2011). A discriminative model for age invariant face recognition. *IEEE transactions on information forensics and security*, 6(3), 1028-1037.

[156] Mun, M., & Deorankar, A. (2014). Implementation of plastic surgery face recognition using multimodal biometric features. *Int. J. Comput. Sci. Inform. Technol*, 5(3), 3711-3715.

[157] Wang, M., & Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*, 429, 215-244.

[158] Zhang, Y., Deng, W., Wang, M., Hu, J., Li, X., Zhao, D., & Wen, D. (2020). Global-local gcn: Large-scale label noise cleansing for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7731-7740).

[159] "Ms-celeb-1m challenge 3". [Online] Available: *http://trillionpairs.deepglint.com*.

[160] Wang, F., Chen, L., Li, C., Huang, S., Chen, Y., Qian, C., & Loy, C. C. (2018). The devil of face recognition is in the noise. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 765-780).

[161] Bansal, A., Castillo, C., Ranjan, R., & Chellappa, R. (2017). The do's and don'ts for cnn-based face verification. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 2545-2554).

[162] Guo, Y., Zhang, L., Hu, Y., He, X., & Gao, J. (2016). Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ComputerVision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14* (pp. 87-102). Springer International Publishing.

[163] Nech, A., & Kemelmacher-Shlizerman, I. (2017). Level playing field for million scale face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7044-7053).

[164] Yi, D., Lei, Z., Liao, S., & Li, S. Z. (2014). Learning face representation from scratch. *arXiv preprint arXiv:1411.7923.*

[165] Maze, B., Adams, J., Duncan, J. A., Kalka, N., Miller, T., Otto, C., ... & Grother, P. (2018, February). Iarpa janus benchmark-c: Face dataset and protocol. In *2018 international conference on biometrics (ICB)* (pp. 158-165). IEEE.

[166] Wang, M., Deng, W., Hu, J., Tao, X., & Huang, Y. (2019). Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 692-702).

[167] Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., ... & Grother, P. (2017). Iarpa janus benchmark-b face dataset. In *proceedings on the IEEE conference on computer vision and pattern recognition workshops* (pp. 90-98).

[168] Zheng, T., Deng, W., & Hu, J. (2017). Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*.

[169] Sengupta, S., Chen, J. C., Castillo, C., Patel, V. M., Chellappa, R., & Jacobs, D. W. (2016, March). Frontal to profile face verification in the wild. In *2016 IEEE winter conference on applications of computer vision (WACV)* (pp. 1-9). IEEE.

[170] Bansal, A., Nanduri, A., Castillo, C. D., Ranjan, R., & Chellappa, R. (2017, October). Umdfaces: An annotated face dataset for training deep networks. In *2017 IEEE international joint conference on biometrics (IJCB)* (pp. 464-473). IEEE.

[171] Klare, B. F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., ... & Jain, A. K. (2015). Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1931-1939).

[172] Beveridge, J. R., Phillips, P. J., Bolme, D. S., Draper, B. A., Givens, G. H., Lui, Y. M., ... & Cheng, S. (2013, September). The challenge of face recognition from digital point-and-shoot cameras. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)* (pp. 1-8). IEEE.

[173] Wong, Y., Chen, S., Mau, S., Sanderson, C., & Lovell, B. C. (2011, June). Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In *CVPR 2011 WORKSHOPS* (pp. 74-81). IEEE.

[174] "Fg-net aging database". [Online]. Available: *http://www.fgnet.rsunit.com*.

[175] Kim, M., Kumar, S., Pavlovic, V., & Rowley, H. (2008, June). Face tracking and recognition with visual constraints in real-world videos. In *2008 IEEE Conference on computer vision and pattern recognition* (pp. 1-8). IEEE.

[176] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1), 1-48.

[177] Lv, J. J., Shao, X. H., Huang, J. S., Zhou, X. D., & Zhou, X. (2017). Data augmentation for face recognition. *Neurocomputing*, 230, 184-196.

[178] Li, B., Cui, Y., Lin, T. Y., & Belongie, S. (2022, October). SITTA: Single Image Texture Translation for Data Augmentation. In *European Conference on Computer Vision* (pp. 3-20). Cham: Springer Nature Switzerland.

[179] Baran, I., Kupyn, O., & Kravchenko, A. (2019). Safe augmentation: Learning task-specific transformations from data. *arXiv preprint arXiv:1907.12896*.

[180] Ma, W. B., Deng, X. Y., Yang, Y., & Fang, W. C. (2022, October). An effective lung sound classification system for respiratory disease diagnosis using densenet cnn model with sound pre-processing engine. In *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (pp. 218-222). IEEE.

[181] Krinski, B. A., Ruiz, D. V., & Todt, E. (2022). Light In The Black: An Evaluation of Data Augmentation Techniques for COVID-19 CT's Semantic Segmentation. *arXiv preprint arXiv:2205.09722*.

[182] Andriyanov, N., & Andriyanov, D. (2020, May). Pattern recognition on radar images using augmentation. In 2020 *Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)* (pp. 0289-0291). IEEE.

[183] Agustin, T., Utami, E., & Al Fatta, H. (2020, November). Implementation of data augmentation to improve performance CNN method for detecting diabetic retinopathy. In *2020 3rd International Conference on Information and Communications Technology (ICOIACT)* (pp. 83-88). IEEE.

[184] Zeiser, F. A., da Costa, C. A., Zonta, T., Marques, N. M., Roehe, A. V., Moreno, M., & da Rosa Righi, R. (2020). Segmentation of masses on mammograms using data augmentation and deep learning. *Journal of digital imaging*, 33, 858-868.

[185] Henna, S., & Reji, A. (2021). A data augmented approach to transfer learning for Covid-19 detection. *arXiv preprint arXiv:2108.02870*.

[186] Ottom, M. A., Rahman, H. A., & Dinov, I. D. (2022). Znet: deep learning approach for 2D MRI brain tumor segmentation. *IEEE Journal of Translational Engineering in Health and Medicine*, 10, 1-8.

[187] Shijie, J., Ping, W., Peiyi, J., & Siping, H. (2017, October). Research on data augmentation for image classification based on convolution neural networks. In *2017 Chinese automation congress (CAC)* (pp. 4165-4170). IEEE.

[188] Zheng, X., Chalasani, T., Ghosal, K., Lutz, S., & Smolic, A. (2019). Stada: Style transfer as data augmentation. *arXiv preprint arXiv:1909.01056*.

[189] Monshi, M. M. A., Poon, J., Chung, V., & Monshi, F. M. (2021). CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR. *Computers in biology and medicine*, 133, 104375.

[190] Budhiman, A., Suyanto, S., & Arifianto, A. (2019, December). Melanoma cancer classification using resnet with data augmentation. In 2019 *International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)* (pp. 17-20). IEEE.

[191] Aleem, S., Kumar, T., Little, S., Bendechache, M., Brennan, R., & McGuinness, K. (2022). Random data augmentation based enhancement: a generalized enhancement approach for medical datasets. *arXiv preprint arXiv:2210.00824*.

[192] Hsu, C. Y., Lin, L. E., & Lin, C. H. (2021). Age and gender recognition with random occluded data augmentation on facial images. *Multimedia Tools and Applications*, 80, 11631-11653.

[193] Scott, G. J., England, M. R., Starms, W. A., Marcum, R. A., & Davis, C. H. (2017). Training deep convolutional neural networks for land–cover classification of high-resolution imagery. *IEEE Geoscience and Remote Sensing Letters*, 14(4), 549-553.

[194] Yuan, J., Liu, Y., Shen, C., Wang, Z., & Li, H. (2021). A simple baseline for semi-supervised semantic segmentation with strong data augmentation.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 8229-8238).

[195] Prakhar. Haversine formula to find distance between two points on a sphere (2020). [Online]. Available: *https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/.* [Accessed 12 July 2023].

[196] Morissette, L., & Chartier, S. (2013). The k-means clustering technique: General considerations and implementation in Mathematica. *Tutorials in Quantitative Methods for Psychology*, 9(1), 15-24.

[197] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026-1034).

[198] Glorot, X. and Bengio, Y., 2010, March. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256). JMLRWorkshop and Conference Proceedings.

[199] Garbin, C., Zhu, X., & Marques, O. (2020). Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia Tools and Applications*, 79, 12777-12815.

[200] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807-814).

[201] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (Vol. 30, No. 1, p. 3).

[202] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02), 107-116.

[203] Mascarenhas, S., & Agarwal, M. (2021, November). A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)* (Vol. 1, pp. 96-99). IEEE.

[204] Ikechukwu, A.V., Murali, S., Deepu, R., & Shivamurthy, R. C. (2021). ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. *Global Transitions Proceedings*, 2(2), 375-381.

[205] Vulli, A., Srinivasu, P. N., Sashank, M. S. K., Shafi, J., Choi, J., & Ijaz, M. F. (2022). Fine-tuned DenseNet-169 for breast cancer metastasis prediction using FastAI and 1-cycle policy. *Sensors*, 22(8), 2988.

[206] Brownlee, J. (2021). How to Develop a Weighted Average Ensemble With Python. [Online]. Available: *https://machinelearningmastery.com/weighted-average-ensemblewith-python*/. [Accessed 15 July 2023].

[207] Venu, S. K. (2020). An ensemble-based approach by fine-tuning the deep transfer learning models to classify pneumonia from chest X-ray images. *arXiv preprint arXiv:2011.05543*.

[208] Smith, L. N. (2017, March). Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)* (pp. 464-472). IEEE.

[209] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[210] Wang, K., Chen, C., & He, Y. (2020, April). Research on pig face recognition model based on keras convolutional neural network. In *IOP Conference Series: Earth and Environmental Science* (Vol. 474, No. 3, p. 032030). IOP Publishing.

[211] Winston, J. J., Hemanth, D. J., Angelopoulou, A., & Kapetanios, E. (2022). Hybrid deep convolutional neural models for iris image recognition. *Multimedia Tools and Applications*, 1-23.

[212] Sreekala, K., Cyril, C. P. D., Neelakandan, S., Chandrasekaran, S., Walia, R., & Martinson, E. O. (2022). Capsule network-based deep transfer learning model for face recognition. *Wireless Communications and Mobile Computing*, 1-12.

[213] Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3), 107-115.

[214] Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

[215] Rosebrock, A. (2019). Keras Learning Rate Finder. [Online] Available: *https://pyimagesearch.com/2019/08/05/keras-learning-rate-finder/*. [Accessed 18 July 2023].

[216] Overview of hyperparameter tuning | Vertex AI | Google Cloud. (n.d.). Google Cloud. [Online]. Available: *https://cloud.google.com/vertex-ai/docs/training/hyperparameter-tuning-overview*. [Accessed 15 August 2023].

[217] Anyscale - What is hyperparameter tuning? (n.d.). Anyscale. [Online] Available: *https://www.anyscale.com/blog/what-is-hyperparameter-tuning.* [Accessed 20 August 2023].

[218] Kumar, A. (2020). Performance Metrics for Machine Learning Models. [Online]. Available: *https://medium.com/analytics-vidhya/performance-metrics-for-machinelearning-models48990018ebd6.* [Accessed 22 August 2023].

[219] Afonja, T. (2017). Accuracy Paradox. [Online]. Available: *https://towardsdatascience.com/accuracy-paradox-897a69e2dd9b*. [Accessed 22 August 2023].

[220] Fan, J., Upadhye, S., & Worster, A. (2006). Understanding receiver operating characteristic (ROC) curves. *Canadian Journal of Emergency Medicine*, 8(1), 19-20.

[221] Team, S. (2017). Face detection using OpenCV and Python: A beginner's guide. [Online]. Available: *https://www.superdatascience.com/blogs/opencv-face-detection/*. [Accessed 27 August 2023].

[222] Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

[223] Mishra, N. K., & Singh, S. K. (2022). Regularized Hardmining loss for face recognition. *Image and Vision Computing*, 117, 104343.

[224] Ben Fredj, H., Bouguezzi, S., & Souani, C. (2021). Face recognition in unconstrained environment with CNN. *The Visual Computer*, 37, 217-226.

[225] Kang, K. (2019). Comparison of face recognition and detection models: Using different convolution neural networks. *Optical Memory and Neural Networks*, 28, 101-108.

[226] Liu, W., Zhou, L., & Chen, J. (2021). Face recognition based on lightweight convolutional neural networks. *Information*, 12(5), 191.

[227] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 212-220).

[228] Zhang, Y., Liu, W., Fan, H., Zou, Y., Cui, Z., & Wang, Q. (2022). Dictionary learning and face recognition based on sample expansion. *Applied Intelligence*, 1-15.

[229] Muqeet, M. A., & Holambe, R. S. (2019). Local binary patterns based on directional wavelet transform for expression and pose-invariant face recognition. *Applied Computing and Informatics*, 15(2), 163-171.

[230] Ayyad, M., & Khalid, C. (2019). New fusion of SVD and Relevance Weighted LDA for face recognition. *Procedia computer science*, 148, 380-388.

[231] Dora, L., Agrawal, S., Panda, R., & Abraham, A. (2017). An evolutionary single Gabor kernel based filter approach to face recognition. *Engineering Applications of Artificial Intelligence*, 62, 286-301.

[232] Liu, R., & Tan, W. (2021). Eqface: A simple explicit quality network for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1482-1490).

[233] Meyes, R., Lu, M., de Puiseau, C. W., & Meisen, T. (2019). Ablation studies in artificial neural networks. *arXiv preprint arXiv:1901.08644*.

[234] Antoniadis, P. (2017). Accuracy Paradox. [Online]. Available: *https://www.baeldung.com/cs/ml-ablation-study*. [Accessed 29 August 2023].

[235] Anwarul, S., Choudhury, T., Dahiya, S. (2023). Dataset of Mugshots. [Online]. Available: *https://data.mendeley.com/datasets/226275vfxz/2*. Mendeley Data, V2.

LIST OF PUBLICATIONS

1. Shahina Anwarul, Susheela Dahiya. "*A Comprehensive Review on Face Recognition Methods and Factors Affecting Facial Recognition Accuracy* ". Proceedings of ICRIC 2019. Lecture Notes in Electrical Engineering, vol 597. Springer, Cham. https://doi.org/10.1007/978-3-030-29407-6_36

2. Shahina Anwarul, Susheela Dahiya. "*Rectified DenseNet169-based automated criminal recognition system for the prediction of crime prone areas using face recognition* ". Journal of Electronic Imaging 31(4), 043055 (30 August 2022). https://doi.org/10.1117/1.JEI.31.4.043055

3. Shahina Anwarul, Susheela Dahiya, Tanupriya Choudhury. *"Performance Analysis of Deep Learning based Face Recognition Model for Video Surveillance,"* 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2022, pp. 1439-1445, https://doi: 10.1109/SMART55829.2022.10047078

4. Shahina Anwarul, Susheela Dahiya, Tanupriya Choudhury. "*A Novel Hybrid Ensemble Convolutional Neural Network (HE-CNN) for Face Recognition by Optimizing Hyperparameters*". Nonlinear Engineering. Modeling and Application (2023). doi: https://doi.org/10.1515/nleng-2022-0290

5. Shahina Anwarul, Tanupriya Choudhury, Susheela Dahiya. "*Dataset of Mugshots*". Mendeley Data (2023), V2, doi: 10.17632/226275vfxz.2

Thesis_Plag_Report

ORIGINALITY REPORT

9%	5%	9%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dokumen.pub Internet Source	1%
2	espace.etsmtl.ca Internet Source	1%
3	link.springer.com Internet Source	<1%
4	"Pattern Recognition and Computer Vision", Springer Science and Business Media LLC, 2019 Publication	<1%
5	Lecture Notes in Computer Science, 2015. Publication	<1%
6	ebin.pub Internet Source	<1%
7	"Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery", Springer Science and Business Media LLC, 2022 Publication	<1%
8	"Computer Vision – ECCV 2020 Workshops", Springer Science and Business Media LLC,	<1%