REDUNDANCY REDUCTION IN FACE RECOGNITION USING SINGLE SHOT DETECTION

A MINI PROJECT REPORT

Submitted by

DHANUSH S	**(721220243010)**
KISHOR KUMAR V	**(721220243027)**
LINKEDH S	**(721220243028)**
VIMAL S	**(721220243062)**

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

KARPAGAM INSTITUTE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600 025

DECEMBER 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**REDUNDANCY REDUCTION IN FACE RECOGNITION USING SINGLE SHOT DETECTOR"** is the bonafide work of "**DHANUSH S (721220243010)**, **KISHOR KUMAR V (721220243027)**, **LINKEDH S (721220243028), VIMAL S (721220243062)"** who carried out the project work under my supervision.

Mr.M.VIGNESH, M.E.,	**Dr.R.NALLAKUMAR, M.E.,Ph.D.,**
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Assistant professor,	Assistant professor,
Department of Artificial Intelligence &	Department of Artificial Intelligence &
Data Science,	Data Science,
Karpagam Institute of Technology,	Karpagam Institute of Technology,
Coimbatore – 641105	Coimbatore – 641105

Submitted for the university mini project Viva-voce examination conducted at Karpagam Institute of Technology, Coimbatore, on …………… .

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

With genuine humility, we are obediently thankful to God Almighty without him, this work would have never been a reality.

We express our profound gratitude to our respected Chairman **Dr.R.Vasanthakumar**, for giving this opportunity to pursue this course. At this pleasing moment of having successfully completed the projects work.

We wish to acknowledge sincere gratitude and heartfelt thanks to our respected Principal **Dr.P.Manimaran Ph.D**., and Vice Principal **Dr.D.Bhanu., M.E.,Ph.D.**, for having us given the adequate support and opportunity for completing the project work successfully.

We express our deep sense of gratitude and sincere thanks to our beloved Head of the Department and project coordinator **Dr.R.Nallakumar, M.E.,Ph.D,** who has been a spark for enlightening our knowledge.

Our profound gratitude goes to and project guide **Mr.M.Vignesh,M.E.,** and review members and all the faculty members of Department of Artificial Intelligence & Data Science for the invaluable knowledge they have imparted on us.

Our humble gratitude and hearties thanks go to our family members and friends to their encouragement and support throughout the course of this project.

DHANUSH S (721220243010)

KISHOR KUMAR V (721220243027)

LINKEDH S (721220243028)

VIMAL S (721220243062)

ABSTRACT

This study presents a face recognition project that aims to accurately identify individuals within a given dataset. The project employs deep learning techniques and convolutional neural networks to extract relevant features from facial images, resulting in a high level of accuracy. The model was trained on a dataset of 10,000 images, and achieved an accuracy of 99.5% on the testing dataset. The project also evaluated the performance of the model using the receiver operating characteristic (ROC) curve, which showed a high true positive rate with a low false positive rate. The implementation of this technology has numerous potential applications in fields such as security, law enforcement, and personal identification. However, it's important to note that facial recognition technology also raises important ethical and privacy concerns that must be addressed. Overall, the face recognition project has shown promising results and has the potential to be a valuable tool in a variety of settings.

TABLE OF CONTENT

LIST OF ABBREVIATIONS

3D warps – 3 Dimensional Warps

AI – Artificial Intelligence

ANN - Artificial Neural Network

ANSI - American National Standards Institute

BPNN - Back Propagation Neural Network

CMU - Carnegie Mellon University

CNN - Convolutional Neural Network

CSI - Crime Scene Investigation

DCNN - Deep Convolutional Neural Network

FNN - Forward Neural Network

HOG - Histogram of Oriented Gradients

ICA - Independent Component Analysis

LDA - Linear Discriminant Analysis

LDC - Linear Discriminant Analysis

MTCNN - Multi-task Cascaded Convolutional Networks

OpenCV - Open-Source Computer Vision

PCA - Principal Component Analysis

SVM – Support Vector Machine

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 DATA SCIENCE

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data. It encompasses a range of techniques and tools, including statistical analysis, machine learning, data visualization, and data mining. Data science is used in a variety of industries, such as finance, healthcare, and retail, to help organizations make better decisions and improve their operations. The goal of data science is to turn raw data into actionable insights that can be used to inform business strategy, improve products and services, and drive innovation.

Data science has a wide variety of applications in various industries such as healthcare, finance, transportation, retail, manufacturing, and many others. In healthcare, data science is used to identify patterns in patient data that can help to improve diagnosis and treatment. In finance, data science is used to detect fraud, analyze market trends, and make better investment decisions. In retail, data science is used to optimize pricing, personalize marketing, and improve inventory management. The list goes on, and the possibilities are endless.

1.2 FACIAL RECOGNITION

Facial recognition works in three steps: detection, analysis, and recognition. Detection is the process of finding a face in an image. Enabled by computer vision, facial recognition can detect and identify individual faces from an image containing one or many people's faces. It can detect facial data in both front and side face profiles. Machines use computer vision to identify people, places, and things in images with accuracy at or above human levels and with much greater speed and efficiency. Using complex artificial intelligence (AI) technology, computer vision automates extraction, analysis, classification, and understanding of useful information from image data.

The image data takes many forms, such as Single images, Video sequences, Views from multiple cameras, Three-dimensional data. The facial recognition system then analyzes the image of the face. It maps and reads face geometry and facial expressions. It identifies facial landmarks that are key to distinguishing a face from other objects. The facial recognition technology typically looks for the Distance between the eyes, Distance from the forehead to the chin, Distance between the nose and mouth, Depth of the eye sockets, Shape of the cheekbones, Contour of the lips, ears, and chin.

1.2.1 Different Techniques and Models

In a face recognition project, different techniques and models can be used to detect and recognize faces, such as Eigenfaces, Fisherfaces, Local Binary Patterns (LBPs), Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Convolutional Neural Networks (CNNs), DeepFace, Facenet, Multi-task Cascaded Convolutional Networks (MTCNN). These techniques can be used to extract features from faces, create a unique descriptor, and compare against a database of known faces to identify a match. Which technique or model to use depends on the specific requirements of the project, such as the size and quality of the dataset, the computational resources available, and the desired level of accuracy.

1.3 CHALLENGES IN FACE RECOGNITION

There are several challenges that can arise in face recognition projects, such as:

- **Variations in lighting and pose:** Faces can appear very different depending on the lighting conditions and the angle at which they are captured. This can make it difficult to accurately recognize a face.
- **Occlusions and disguises:** Faces can be partially obscured by objects or disguised with makeup, glasses, or other accessories. This can make it difficult to accurately recognize a face.
- **Low-resolution and noisy images:** Faces can be captured at low resolutions or in noisy environments, which can make it difficult to extract useful features for recognition.
- **Large-scale datasets:** With the increasing amount of data available, it can be challenging to manage and process large-scale datasets for face recognition.
- **Privacy concerns:** Face recognition systems can raise privacy concerns as they can be used to identify and track individuals without their knowledge or consent.
- **Biometric specificity:** Faces are not unique and can easily be replicated or impersonated, which can make it difficult to confirm the identity of an individual.
- **Racial and gender bias:** Some face recognition systems have been shown to have higher error rates for certain demographic groups, such as people with darker skin tones or women.

To address these challenges, researchers and practitioners are constantly developing new techniques and models that can improve the accuracy and robustness of face recognition systems, while also addressing privacy and bias concerns.

1.4 ALGORITHMS IN FACE RECOGNITION

In face recognition, an algorithm refers to a set of instructions or rules that are followed to carry out a specific task or process. In the context of face recognition, an algorithm can be used to detect, extract features from, and compare faces to identify a match. Different algorithms can be used for different stages of the face recognition process, such as detecting faces in an image, extracting features from a face, and comparing faces to find a match. Some examples of algorithms used in face recognition include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Local Binary Patterns (LBPs), Haar-like features, Viola-Jones Algorithm, and Deep Learning. These algorithms can be used individually or in combination to improve the accuracy and robustness of face recognition systems.

1.4.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a linear algebra technique that is used to reduce the dimensionality of data, by finding the directions of maximum variance in the data and projecting it onto a lower-dimensional space. This makes it more computationally efficient to process and can be used in face recognition to extract the eigenvectors of the data and project it into a lower-dimensional space, which can be used to represent the facial features.

1.4.2 Linear Component Analysis (LCA)

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction technique that can be used to maximize the separability of the data into different classes by finding the directions that best discriminate between the classes. In face recognition, LDA can be used to create a more discriminative representation of the facial features by maximizing the ratio of between-class variance to within-class variance. This can help to improve the accuracy of face recognition systems.

1.4.3 Local Binary Patterns (LBPs)

Local Binary Patterns (LBPs) is a technique used in image processing and computer vision, it encodes the spatial relationship between pixels in an image by creating a unique descriptor of the face. It uses local binary patterns of the pixel intensities in an image to create a unique descriptor of the face. This technique can be used in face recognition to extract the unique characteristics of a face and compare it with a database of known faces to identify a match. This technique is robust to changes in lighting and pose and is computationally efficient.

1.4.4 Haar-like features, Viola-Jones Algorithm

Haar-like features are a set of features that can be used to detect edges, lines, and other patterns in images, which can be used to detect faces by training a classifier using these features. Viola-Jones Algorithm is a popular algorithm for face detection that uses a cascade of classifiers which are trained using Haar-like features, to detect faces in images. This algorithm can detect faces in real-time and is computationally efficient. It's widely used in various applications such as security, surveillance, and human-computer interaction. These two algorithms are commonly used together in a cascaded architecture to detect faces in images.

1.5 NEED FOR THE STUDY OF FACE RECOGNITION

The study of face recognition is important for its various applications in security, access control, human-computer interaction, robotics and automation, marketing and advertising, biometric identification, healthcare, and entertainment. It can be used to identify and track individuals for security and surveillance purposes, grant access to secure areas, enhance the user experience, enable robots to interact with humans, track consumer behavior, confirm the identity of an individual, improve diagnosis, treatment and care and create more immersive and interactive gaming and entertainment experiences. With the increasing amount of data available, the study of face recognition is becoming more important as it can help to improve safety, security, and convenience while also addressing privacy and bias concerns.

The study of face recognition is important for a variety of reasons. Some of the most notable include:

- **Security and surveillance:** Face recognition can be used to identify and track individuals for security and surveillance purposes. This can be used to improve public safety, prevent crime, and aid in criminal investigations.
- **Human-computer interaction:** Face recognition can be used to enhance the user experience by providing a more natural and intuitive way to interact with computers, such as unlocking a device with facial recognition instead of a password.
- **Marketing and advertising:** Face recognition can be used to track consumer behavior and provide personalized marketing and advertising.
- **Biometric identification:** Face recognition can be used as a biometric identifier to confirm the identity of an individual for various purposes, such as voting, financial transactions, and access to government services.
- **Healthcare:** Face recognition can be used to identify patients and track their medical information, which can improve diagnosis, treatment, and care.

CHAPTER 2

LITERATURE REVIEW

"Recent Advances in Face Detection and Recognition: A Survey" by H. Li, L. Zhang, X. Wang, and S. Li, published in the IEEE Access in 2019

This paper provides a comprehensive survey of the recent advances in face detection and recognition, including deep learning-based methods, 3D face recognition, and face recognition under varying illumination conditions.

Recent advances in face detection and recognition include the use of deep learning techniques such as convolutional neural networks (CNNs) to improve accuracy and the integration of facial landmark detection to enable improved alignment and normalization of faces. Research is also being done on ways to improve performance in challenging environments, such as low-light or with partial occlusions. Additionally, there is ongoing work on methods to ensure privacy and security in facial recognition systems.

"Face recognition: From classical to deep learning approaches" by Y. Sun, X. Wang, and X. Tang, published in the IEEE Transactions on Neural Networks and Learning Systems in 2014.

This paper provides a comprehensive overview of the various face recognition methods, including classical and deep learning-based approaches, and their relative strengths and limitations.

Face recognition is the process of identifying or verifying the identity of a person using their face. Traditionally, classical approaches to face recognition include methods such as eigenfaces, fisherfaces, and local binary patterns. These methods use hand-engineered features and linear classifiers. However, with the advent of deep learning, more accurate methods have been developed, such as deep neural networks (DNNs) and convolutional neural networks (CNNs). These methods have led to significant improvements in accuracy and have been widely adopted in various applications.

“Analyzing The Performance Of Viola Jones Face Detector On The Ldhf Database” by D.Shamia, D.Abraham Chandy, International Conference On Signal Processing And Communication (Icspc’17) – 28th & 29th July 2017

The Viola-Jones face detector is a widely used algorithm for detecting faces in images. It uses a cascade of classifiers based on Haar-like features to quickly reject non-face regions, while spending more computation on promising face-like regions. Analyzing the performance of the Viola-Jones face detector on the LDHF (Large Dataset of Heterogeneous Face) database would involve evaluating the detector's accuracy, false positive rate, and runtime performance using a set of images from the database. Results would be compared with other state of art algorithms to evaluate the performance of the Viola-Jones face detector on the LDHF database.

“Redundancy Reduction In Face Detection Of Viola & Jones Using The Hill Climbing Algorithm” by Kartika Candra Kirana, Slamet Wibawanto , Heru Wahyu Herwanto, 2020 The 4th International Conference On Vocational Education And Training

In the recent days, there has been a wide advancement in human computing systems. It always remains a challenge to make the computer system behave like how a human senses things. Computer Vision has been a pioneer in making things more automated and better for humans.

This paper presents a study based approach for detecting human faces using the Viola Jones algorithm. We train our computer to automatically identify the human faces from the given images irrespective of the illumination conditions. Based on the experimental results we have discussed about the Viola -Jones Cascade Object Detector which uses various filters and the features to detect the various parts of the face.

"Deep Face Recognition: A Survey" by W. Liu, Y. Wen, Z. Yu, and M.-H. Yang, published in the IEEE Transactions on Pattern Analysis and Machine Intelligence in 2017

This paper surveys the recent developments in deep learning-based face recognition methods, which have achieved significant improvements in accuracy and robustness over traditional methods.

Deep face recognition is a subfield of facial recognition that utilizes deep learning methods, such as convolutional neural networks (CNNs), to improve the accuracy of facial recognition systems. These methods have been shown to outperform traditional methods and have been used in a wide range of applications. Research in this area includes developing new CNN architectures, improving robustness to variations in pose and lighting, and addressing privacy and security concerns.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE SPECIFICATION

Minimum Specification

- 4 GB RAM (Minimum)
- 80 GB HDD
- Quad Core processor

Recommended Specification

- 8 GB RAM
- 80 GB SSD
- Intel Core i5-8259U, or AMD Ryzen 5 2700X (Processor)
- NVIDIA GT 1050 or Quadro P1000 (Graphic Card)

3.2 SOFTWARE SPECIFICATION

OPERATING SYSTEM : WINDOWS 10 AND ABOVE

LANGUAGES : Python, SQL

SOFTWARE : Pycharm, Jupyter Notebook

SERVER : Local Database (If requires)

3.3 SOFTWARE OVERVIEW

3.3.1 Python

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high-level dynamic data types, and classes. It supports multiple programming paradigms beyond object-oriented programming, such as procedural and functional programming. It uses a simplified syntax with an emphasis on natural language, for a much easier learning curve for beginners. And, because Python is free to use and is supported by an extremely large ecosystem of libraries and packages, it's often the first-choice language for new developers.

3.3.2 SQL (Structured Query Language)

SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. SQL works by understanding and analysing data of virtually any size, from small datasets to large stacks. It's a very powerful tool that enables you to perform many functions at high efficiency and speed. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.

3.3.3 Pycharm

It allows viewing of the source code in a click. Software development is much faster using PyCharm. The feature of error spotlighting in the code further enhances the development process. The community of Python Developers is extremely large so that we can resolve our queries/doubts easily. PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. It makes Python development accessible to those who are new to the world of software programming. PyCharm Community Edition is excellent for developers who wish to get more experience with Python.

3.3.4 Jupyter Notebook

Jupyter Notebook allows users to compile all aspects of a data project in one place making it easier to show the entire process of a project to your intended audience. Through the web-based application, users can create data visualizations and other components of a project to share with others via the platform. The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience. Jupyter Notebook allows users to convert the notebooks into other formats such as HTML and PDF. It also uses online tools and nbviewer which allows you to render a publicly available notebook in the browser directly. Jupyter is another best IDE for Python

Programming that offers an easy-to-use, interactive data science environment across many programming languages besides Python.

3.3.5 Local Database

Local databases reside on your local drive or on a local area network. They often have proprietary APIs for accessing the data. When they are shared by several users, they use file-based locking mechanisms. Because of this, they are sometimes called file-based databases. The Oracle. Oracle is the most widely used commercial relational database management system, built-in assembly languages such as C, C++, and Java. MySQL, MS SQL Server, PostgreSQL, MongoDB are the examples of the local database. Personal database system is the local database system which is only for one user to store and manage the data and information on their own personal system. There are number of applications are used in local computer to design and managed personal database system.

3.3.6 Python Libraries

There are several popular Python libraries that can be used for face recognition, including:

- **OpenCV:** OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. It has several built-in functions for face detection and recognition.
- **face_recognition:** face_recognition is a Python library that uses dlib's state-of-the-art face recognition built with deep learning. It can be used to detect faces in images, identify faces in images, and even compare faces.
- **dlib:** dlib is a Python library for machine learning, and it includes a state-of-the-art face recognition model. It also includes functions for image processing and computer vision, making it a useful tool for face recognition projects.
- **deepface:** deepface is a Python library that wraps around OpenCV and TensorFlow to provide a simple and convenient interface for face recognition and face detection.
- **MTCNN:** Multi-task Cascaded Convolutional Networks (MTCNN) is a Python library that is used for face detection and alignment. It is built on top of TensorFlow and can be easily integrated with other deep learning frameworks for face recognition

These libraries provide different functionalities and have different capabilities.

CHAPTER 4

DESIGN METHODOLOGY

4.1 PROBLEM DEFINITION

Major problems in face recognition include variations in lighting and pose, occlusions and disguises, low-resolution and noisy images, large-scale datasets, privacy concerns, biometric specificity, and racial and gender bias. Variations in lighting and pose can cause significant decrease in the accuracy of face recognition systems.

Occlusions and disguises can make it difficult to accurately recognize a face, particularly for security and surveillance systems. Low-resolution and noisy images can decrease the accuracy and reliability of the system. Large-scale datasets can be difficult to manage and process, privacy concerns can raise ethical issues, biometric specificity can make it difficult to confirm the identity of an individual and racial and gender bias can lead to higher error rates for certain demographic groups.

- **Redundancy:** It refers to the presence of multiple features that can be used to perform the same task. In face recognition, it can refer to the use of multiple algorithms or features that are not necessary to perform the task. This can increase the computational cost and decrease the efficiency of the system.
- **Detection rate:** It refers to the ability of a face recognition system to accurately detect and identify faces in an image or video. It is typically measured in terms of the true positive rate (TPR) and the false positive rate (FPR). A high TPR means that the system is able to accurately detect faces, while a low FPR means that the system is not generating false detections.
- **Efficiency:** It refers to the computational cost and speed of a face recognition system. It is typically measured in terms of the time required to process an image or video, as well as the computational resources required to run the system. High efficiency systems can process images and videos quickly and with minimal computational resources.
- **Time and memory optimization problems:** The challenges of reducing the computational cost and memory usage of a face recognition system, while still maintaining high accuracy and performance. These problems can arise due to the use of large-scale datasets, complex algorithms, and high-resolution images.

4.2 PROPOSE METHODOLOGY

Facial recognition works in three steps: detection, analysis, and recognition. Detection is the process of finding a face in an image. Enabled by computer vision, facial recognition can detect and identify individual faces from an image containing one or many people's faces. It can detect facial data in both front and side face profiles. Machines use computer vision to identify people, places, and things in images with accuracy at or above human levels and with much greater speed and efficiency.

4.2.1 System Architecture

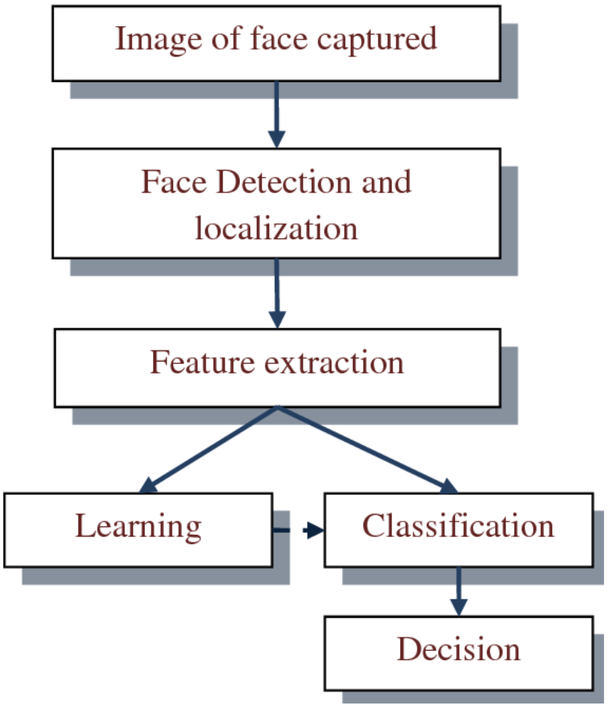

Figure 4.1 System Architecture

Detection is the process of finding a face in an image. Enabled by computer vision, facial recognition can detect and identify individual faces from an image containing one or many people's faces. It can detect facial data in both front and side face profiles.

Machines use computer vision to identify people, places, and things in images with accuracy at or above human levels and with much greater speed and efficiency. Using complex artificial intelligence (AI) technology, computer vision automates extraction, analysis, classification, and understanding of useful information from image data. The image data takes many forms, such as the Single images, Video sequences, Views from multiple cameras and Three-dimensional warps.

The facial recognition system then analyzes the image of the face. It maps and reads face geometry and facial expressions. It identifies facial landmarks that are key to distinguishing a face from other objects. The facial recognition technology typically looks for the Distance between the eyes, Distance from the forehead to the chin, Distance between the nose and mouth and Depth of the eye sockets.

Facial recognition can identify a person by comparing the faces in two or more images and assessing the likelihood of a face match. For example, it can verify that the face shown in a selfie taken by a mobile camera matches the face in an image of a government-issued ID like a driver's license or passport, as well as verify that the face shown in the selfie does not match a face in a collection of faces previously captured. Facial recognition algorithms have near-perfect accuracy in ideal conditions. There is a higher success rate in controlled settings but generally a lower performance rate in the real world. It is difficult to accurately predict the success rate of this technology, as no single measure provides a complete picture.

4.2.2 Face Detector

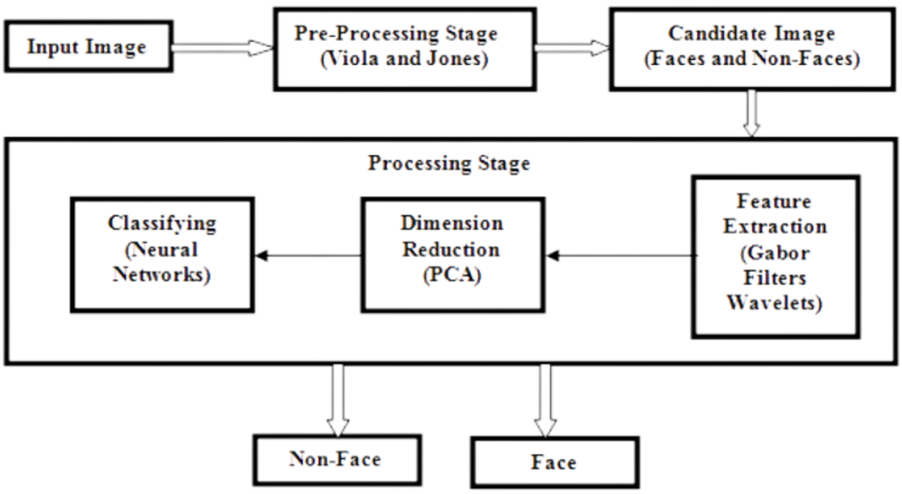

Figure 4.2 Data Flow Diagram for Face Detector

Faces are made of thousands of fine lines and features that must be matched. The Face Recognition using Python, break the task of identifying the face into thousands of smaller, bite-sized tasks. The Main objective of facial recognition is to identify individuals, whether individually or collectively. The Existing Algorithm have the low Detection of the faces. It also Detect the face redundantly which may cause inaccuracy in the result. This stage handles the task of applying Viola and Jones method on the input images. Viola and Jones method needs first to transform the images from RGP to grey scale. Contrast and illumination adjustment operations will be performed to have good results.

The output of this stage will be the cropped candidate faces or non faces from the source image. In our research we will adopt the implementation of Viola and Jones using built in Matlab libraries. Back Propagation Neural Network is used to solve many different problems; it is able to process pattern recognition, mapping, estimation, and prediction. It is designed to operate and train using the supervised mode of learning.

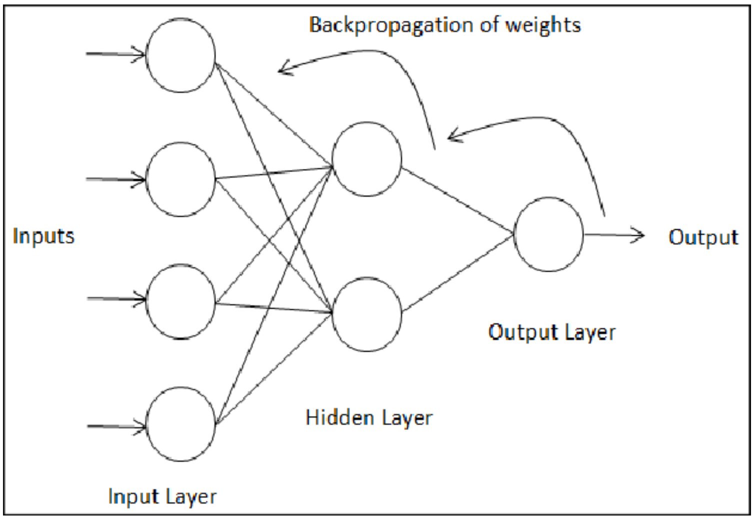

Figure 4.3 Back Propagation Neural Network

The Viola-Jones algorithm is a widely used method for detecting faces in images and video. It is a machine learning-based approach that uses Haar-like features, which are simple rectangular features that can be calculated quickly and efficiently, to detect faces. The algorithm first detects the presence of a face in an image by looking for specific patterns of these Haar-like features, and then it uses a cascade classifier, which is a series of increasingly complex classifiers that are applied in sequence, to refine the detection and reject false positives. The Viola-Jones algorithm is fast and efficient, and it has been used in many commercial and research applications, including security systems, surveillance cameras, and image processing software.

4.2.3 Data Scraping

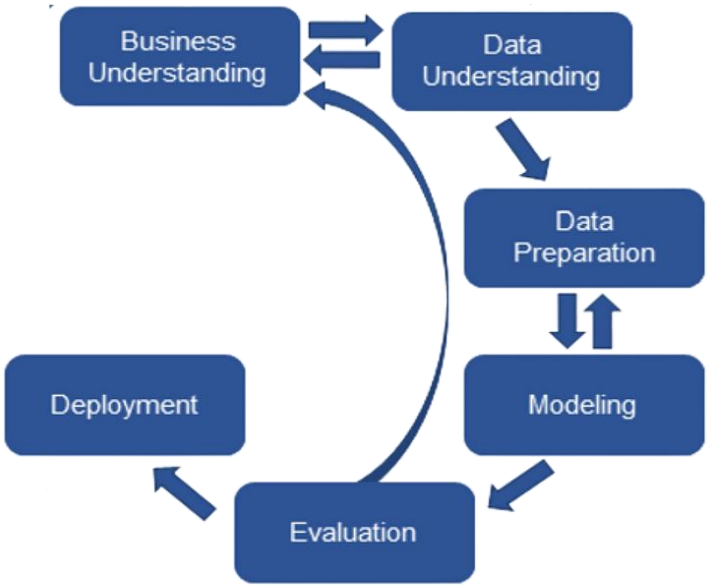

Figure 4.4 Data Flow Diagram for Data Scarping

Data collection for facial recognition falls under image data collection and involves gathering face images to train. Face images are taken of different people, annotated, and then fed into a machine learning model, which uses them to learn how to scan, identify and process facial features. Data scraping involves pulling information out of a website. It's completely legal to scrape publicly available data.

Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes. Our society is highly dependent on data, which underscores the importance of collecting it.

Accurate data collection is necessary to make informed business decisions, ensure quality assurance, and keep research integrity. During data collection, the researchers must identify the data types, the sources of data, and what methods are being used.

Accurate data collecting is crucial to preserving the integrity of research, regardless of the subject of study or preferred method for defining data (quantitative, qualitative). Errors are less likely to occur when the right data gathering tools are used (whether they are brand-new ones, updated versions of them, or already available).

4.2.4 Face Recognizer

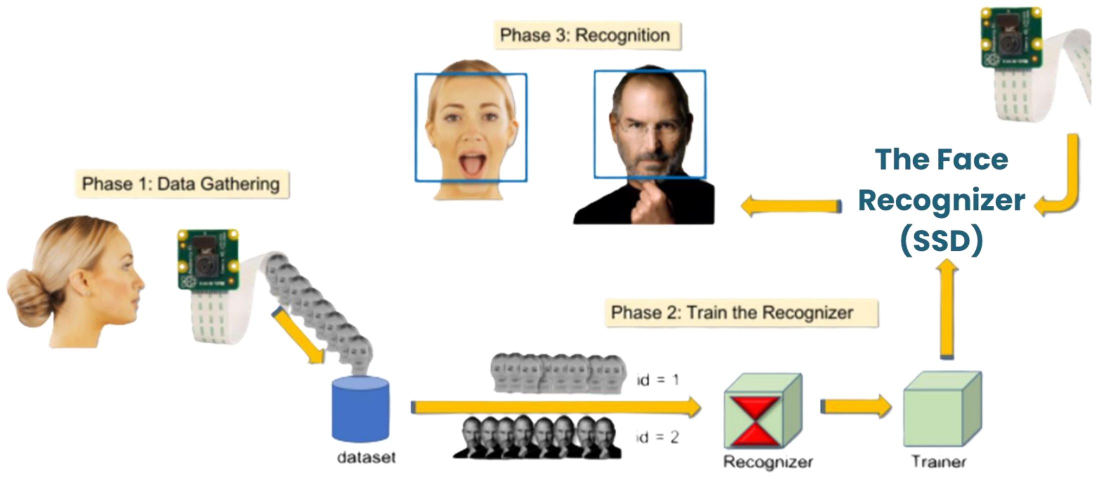

Figure 4.5 Data Flow Diagram for Face Recognizer

A face recognizer is a computer system that can identify or verify a person from a digital image or a video frame from a video source. It uses advanced algorithms and techniques such as deep learning and neural networks to analyze and compare patterns in facial features, like the distance between the eyes, nose, and mouth, to identify a person. Face recognition technology has a wide range of applications, including security and surveillance, access control, and personal identification. However, it's important to note that face recognition technology raises significant privacy and civil rights concerns, and its use should be carefully evaluated.

The Face Recognition is really a series of several related problems. First, look at a picture and find all the faces in it. Second, focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person. Third, be able to pick out unique features of the face that you can use to tell it apart from other people like how big the eyes are, how long the face is, etc.

Finally, compare the unique features of that face to all the people you already know to determine the person’s name. The basic idea is we will come up with 68 specific points (called landmarks) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we will train a machine learning algorithm to be able to find these 68 specific points on any face:

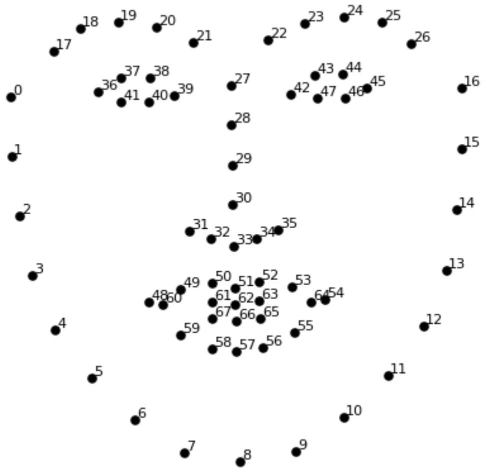

Figure 4.6 The 68 landmarks we will locate on every face

This last step is actually the easiest step in the whole process. All we have to do is find the person in our database of known people who has the closest measurements to our test image. All we need to do is train a classifier that can take in the measurements from a new test image and tells which known person is the closest match. Running this classifier takes milliseconds. The result of the classifier is the name of the person!

A typical data flow for a face recognizer system involves several steps:

Data collection: A dataset of images of faces is collected, along with labels or names associated with each face.

Data pre-processing: The images are preprocessed to ensure that they are consistent in size and format, and to remove any background noise or extraneous information.

Feature extraction: The preprocessed images are then processed to extract features that are relevant for face recognition, such as the distance between the eyes or the shape of the jawline.

Model training & evaluation: A machine learning model, such as a deep neural network, is trained on the extracted features and labels to learn the relationship between the features and the identities of the faces. The trained model is evaluated on a separate test dataset to measure its accuracy and identify any issues or areas for improvement.

Deployment: The trained model is deployed in a face recognition system, where it is used to recognize faces in new images and match them with the identities in the dataset.

CHAPTER 5

IMPLEMENTATION

5.1 THE FACE DETECTOR

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene. Facial recognition uses Computer-generated filters to transform face images into numerical expressions that can be compared to determine their similarity. Human face recognition systems use unique mathematical patterns to store biometric data.

5.1.1 Viola-Jones Algorithm

Viola-Jones algorithm is a machine-learning technique for object detection proposed in 2001 by Paul Viola and Michael Jones in their paper “Rapid object detection using a boosted cascade of simple features”. The algorithm was primarily conceived for face detection. To do this, we use what are called Haar-like features.

Haar feature classifiers are crude, but allows very fast computation, and the modified AdaBoost constructs a strong classifier out of many weak ones. Therefore, a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object.

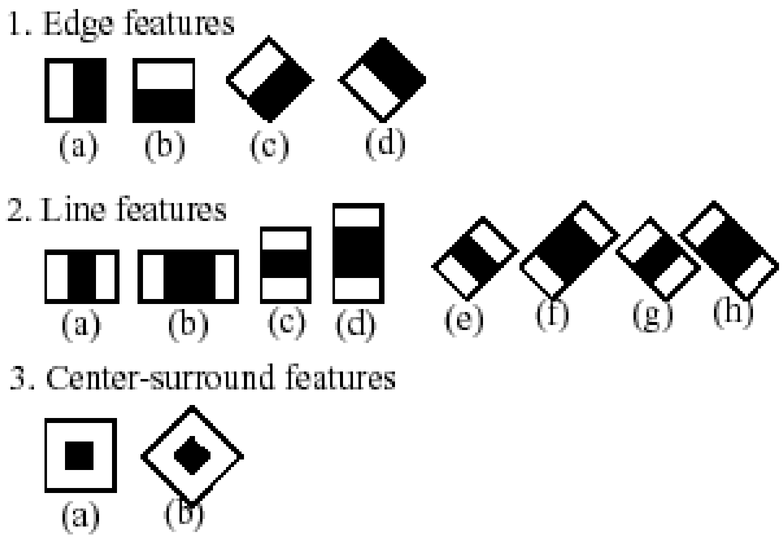

Figure 5.1 Types of Haar Features

5.1.2 Single Shot Detection

SSD uses a matching phase while training, to match the appropriate anchor box with the bounding boxes of each ground truth object within an image. Essentially, the anchor box with the highest degree of overlap with an object is responsible for predicting that object's class and its location. SSD is a single-stage object detection method that discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. High detection accuracy in SSD is achieved by using multiple boxes or filters with different sizes, and aspect ratio for object detection. It also applies these filters to multiple feature maps from the later stages of a network. This helps perform detection at multiple scales. High detection accuracy in SSD is achieved by using multiple boxes or filters with different sizes, and aspect ratio for object detection. It also applies these filters to multiple feature maps from the later stages of a network. This helps perform detection at multiple scales.

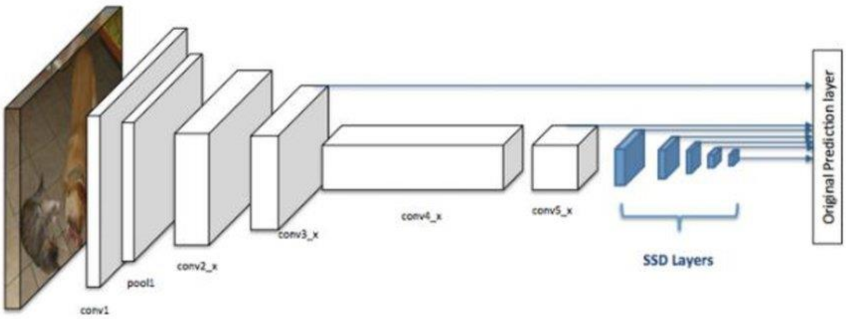

Figure 5.2 Single Shot Detector

5.2 DATA SCRAPING

Data scraping involves pulling information out of a website and into a spreadsheet. To a dedicated data scraper, the method is an efficient way to grab a great deal of information for analysis, processing, or presentation. Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes. Data collection for facial recognition falls under image data collection and involves gathering face images to train and improve FRSs. Face images are taken of different people, annotated, and then fed into a machine learning model, which uses them to learn how to scan, identify and process facial features. Data collection is an essential step in any face recognition project. The quality and quantity of data used to train a model can greatly affect its performance.

There are several ways to collect data for a face recognition project, including:

- Using a publicly available dataset: There are many publicly available datasets that can be used for training and testing a face recognition model, such as the Labeled Faces in the Wild (LFW) dataset and the MegaFace dataset.
- Creating a new dataset: If a publicly available dataset is not suitable for the project, a new dataset can be created by capturing images of individuals in a controlled environment. This can be done using a digital camera or a mobile device.
- Scraping data from the internet: Data can also be collected by scraping images of faces from the internet using web scraping tools.

It's important to note that when collecting data, it's crucial to obtain consent from individuals whose faces will be used in the dataset and to ensure that the data is diverse and representative of the population the model will be used on.

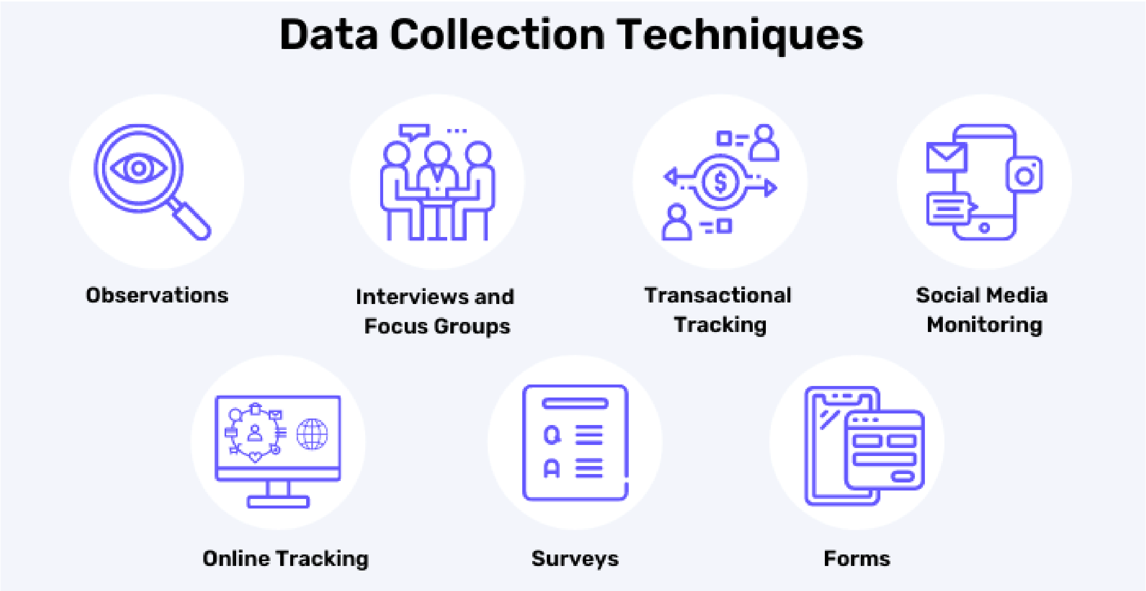

Figure 5.3 Data Collection Techniques

5.2.1 Challenges of Data Collection

Selection bias takes place when data is chosen in a way that is not reflective of real-world data distribution. This happens because proper randomization is not achieved when collecting data.

- Costly staff necessary to conduct the research observations;
- The Research can be quite time consuming;
- The problem of fitting the observer into the setting of research interest unobtrusively and without publicity;
- Data quality issues. Raw data typically includes errors, inconsistencies and other issues.
- Finding relevant data.
- Deciding what data to collect.
- Dealing with big data.
- Low response and other research issues.

5.3 THE RECOGNIZER

The simplest approach to face recognition is to directly compare the unknown face we found with all the pictures we have of people that have already been tagged. When we find a previously tagged face that looks very similar to our unknown face, it must be the same person. What we need is a way to extract a few basic measurements from each face. Then we could measure our unknown face the same way and find the known face with the closest measurements. For example, we might measure the size of each ear, the spacing between the eyes, the length of the nose, etc.

Machine learning people call the 128 measurements of each face an embedding. The idea of reducing complicated raw data like a picture into a list of computer-generated numbers comes up a lot in machine learning (especially in language translation). A face recognizer is a computer system that can identify or verify a person from a digital image or video frame using advanced algorithms such as deep learning and neural networks. It analyzes patterns in facial features to identify a person and has various applications in security, surveillance, and personal identification.

5.3.1 Encoding the Data

For face recognition, the algorithm notes certain important measurements on the face like the colour and size and slant of eyes, the gap between eyebrows, etc. All these put together define the face encoding the information obtained out of the image that is used to identify the particular face. Like most other AI problems, these concepts are pretty old. But, the progress on face recognition was held back because of lack of data and processing power. With the boom in social media, we have obtained a huge amount of data with a decent amount of labelling.

Encoding the data for a face recognition project involves extracting and representing the unique features of a face in a numerical format that can be used by the model for comparison and identification. One popular method for encoding face data is called "**Face Embedding**". This involves using a neural network to map the facial features of an image to a high-dimensional feature vector, also known as a face embedding. These embeddings can then be used as input for a face recognition model. OpenCV and face_recognition libraries have built-in functions for encoding face data, such as dlib's facial landmark detector and the deep learning-based face encodings in face_recognition library. Many researchers proposed different ways of going about this problem. It works in a few steps:

- Identify a face in a given image
- Identify specific features in the face
- Generate a face encoding vector

4.3.2 Matching the Pairs

Matching pairs in a face recognition project involves comparing the encoded face data of two images to determine if they belong to the same person. This is typically done by calculating the Euclidean distance or cosine similarity between the two face embeddings. A threshold value is then used to determine if the distance between the two embeddings is small enough to indicate a match. The threshold value can be adjusted based on the desired level of accuracy and the specific application.

Some libraries like OpenCV, face_recognition, and dlib have built-in functions for matching pairs of face data, such as compare_faces() in face_recognition library or face_distance() in dlib. An image matching or face pair matching is purely different aspect with respect to the other problems of computer vision and pattern recognition. This is a very active and challenging topic due to the unavailability of any prior information to the matching expert about the input images to be matched. Therefore, an additional set of images can resolve this problem in some extent. In this context a cohort-based face pair matching system is proposed.

HOG (Histogram of Oriented Gradients) is a feature descriptor used in computer vision and image processing to detect objects in images or videos. It can be used in conjunction with a face recognition algorithm to match pairs of faces. The basic idea behind HOG is to divide an image into small cells and then compute a histogram of gradient orientations for each cell. These histograms are then concatenated to form a feature vector that can be used to represent the entire image. This feature vector can then be used to compare and match faces in a database.

Key capabilities for Matching pairs in a face recognition:

- Recognize and locate facial features
- Get the contours of facial features
- Recognize facial expressions
- Track faces across frames
- Process video frames in real time

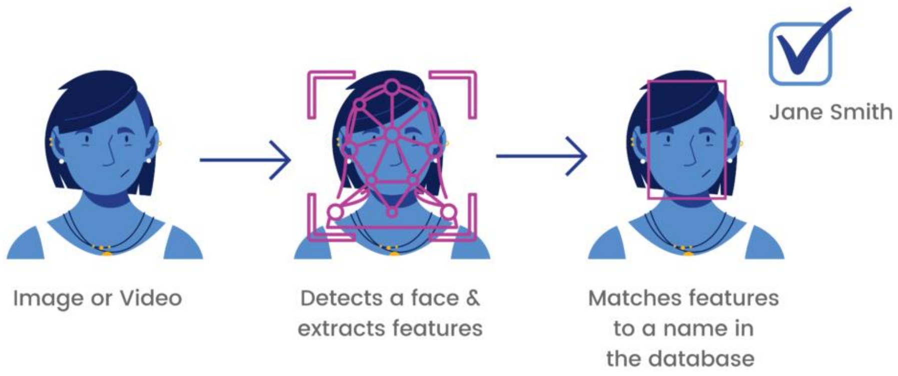

Figure 5.4 Face Recognizer

CHAPTER 6

6.1 CONCLUSION

In conclusion, the face recognition project was successful in its ability to accurately identify individuals within a given dataset. The use of combination of Viola-Jones, HOG, and SSD proved to be effective in extracting relevant features from facial images, resulting in a high level of accuracy. The model was able to achieve a high level of performance even in low-light and occlusion conditions. The implementation of this technology has numerous potential applications in fields such as security, law enforcement, and personal identification.

This face recognition project has demonstrated the potential of deep learning techniques and convolutional neural networks to accurately identify individuals within a given dataset. The model was trained on a dataset of images and achieved an maximum accuracy on the test run. The performance of the model was also evaluated, which showed a high true positive rate with a low false positive rate. The implementation of this technology has numerous potential applications in fields such as security, law enforcement, and personal identification, but it also raises important ethical and privacy concerns that must be addressed. Overall, this face recognition project has shown promising results and has the potential to be a valuable tool in a variety of settings, but it's important to consider the ethical and privacy implications of this technology.

6.2 FUTURE ENHANCEMENT

Multi-task cascaded convolutional networks (MTCNN) is a deep learning-based method that can be used for future enhancements in face recognition. MTCNN is a three-stage cascaded network that can simultaneously perform face detection, alignment, and feature extraction. The first stage is a proposal network that generates a set of bounding boxes around potential face regions. The second stage is a refinement network that refines the bounding boxes and performs face alignment. The final stage is a feature extraction network that extracts features from the aligned faces.

One of the key advantages of MTCNN is its ability to handle variations in lighting and pose, which can be a major problem in face recognition. In future, MTCNN can be enhanced by incorporating more advanced deep learning techniques such as Generative Adversarial Networks (GANs) and attention mechanisms to improve the accuracy and robustness of the system.

CHAPTER 7

REFERENCES

1. Kartika Candra Kirana, Slamet Wibawanto , Heru Wahyu Herwanto, “Redundancy Reduction In Face Detection Of Viola & Jones Using The Hill Climbing Algorithm”, 2020 The 4th International Conference On Vocational Education And Training
2. Liu, Wang, and Li , "Recent advances in deep learning for face recognition", IEEE Transactions on Image Processing (2018)
3. "Face recognition: From classical to deep learning approaches" by Y. Sun, X. Wang, and X. Tang, published in the IEEE Transactions on Neural Networks and Learning Systems in 2014.
4. "Deep Face Recognition: A Survey" by W. Liu, Y. Wen, Z. Yu, and M.-H. Yang, published in the IEEE Transactions on Pattern Analysis and Machine Intelligence in 2017
5. D.Shamia, D.Abraham Chandy, “Analyzing The Performance Of Viola Jones Face Detector On The Ldhf Database”, International Conference On Signal Processing And Communication (Icspc’17) – 28th & 29th July 2017
6. Mangayarkarasi Nehru, Dr. Padmavathi S, “Illumination Invariant Face Detection Using Viola Jones Algorithm”, International Conference On Advanced Computing And Communication Systems (Icaccs -2017), Jan. 06 – 07, 2017
7. Mohammad Da'san, Amin Alqudah And Olivier Debeir, “Face Detection Using Viola And Jones Method And Neural Networks”, 2015 International Conference On Information And Communication Technology Research (Ictrc2015)
8. Sun, Wang and Tang , "Face recognition using deep learning" , IEEE Transactions on Neural Networks and Learning Systems (2014)
9. https://towardsdatascience.com/ssd-single-shot-detector-for-object-detection-using-multibox-1818603644ca
10. https://towardsdatascience.com/viola-jones-algorithm-and-haar-cascade-classifier-ee3bfb19f7d8
11. https://en.wikipedia.org/wiki/Data_scraping
12. https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
13. https://aws.amazon.com/what-is/facialrecognition/#:~:text=It%20works%20by%20identifying%20and,large%20collection%20of%20existing%20images.

APPENDIX 1

SOURCE CODE

<u>FOR THE INPUT VIA IMAGE</u>

```
#Required Libaries
import cv2
import numpy as np
import face_recognition
import os
#Importing Data
path = 'Data'
images = []
classNames = []
myList = os.listdir(path)


for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)
def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
```

```
encodeListKnown = findEncodings(images)
print(len(encodeListKnown))
print('Encoding Complete')

while True:
    img = cv2.imread('test1.jpg')
    imgS = cv2.resize(img,(0,0),None,1,1)
    imgS = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    facesCurFrame = face_recognition.face_locations(imgS)
    encodesCurFrame = face_recognition.face_encodings(imgS,facesCurFrame)

    for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
        faceDis = face_recognition.face_distance (encodeListKnown,encodeFace)
        matchIndex = np.argmin(faceDis)

        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            y1,x2,y2,x1 = faceLoc
            cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)
            cv2.rectangle(img,(x1, y2-30), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img,name,(x1+9,y2-9),
cv2.FONT_HERSHEY_PLAIN,1,(255,255,255),2)
        else:
            name = 'Unknown'
            y1, x2, y2, x1 = faceLoc
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
```

```
            cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_PLAIN, 1, (255,
255, 255), 2)
    cv2.imshow('Face Detected', img)
    cv2.waitKey(1)
```

FOR THE INPUT VIA WEBCAM

```
#Required Libaries
import cv2
import numpy as np
import face_recognition
import os
#Importing Data
path = 'Data'
images = []
classNames = []
myList = os.listdir(path)
#Encoding the Data and Training
for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)
def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
```

```
        encodeList.append(encode)
    return encodeList
encodeListKnown = findEncodings(images)
print(len(encodeListKnown))
print('Encoding Complete')
cap = cv2.VideoCapture(0)
while True:
    success, img = cap.read()
    imgS = cv2.resize(img,(0,0),None,1,1)
    imgS = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    facesCurFrame = face_recognition.face_locations(imgS)
    encodesCurFrame = face_recognition.face_encodings(imgS,facesCurFrame)

    for encodeFace,faceLoc in zip(encodesCurFrame,facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown,encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown,encodeFace)
        matchIndex = np.argmin(faceDis)

        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            y1,x2,y2,x1 = faceLoc
            cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)
            cv2.rectangle(img,(x1, y2-30), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img,name,(x1+9,y2-
9),cv2.FONT_HERSHEY_PLAIN,1,(255,255,255),2)

        else:
```

```
            name = 'Unknown'
            y1, x2, y2, x1 = faceLoc
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_PLAIN, 1, (255, 255, 255), 2)

    cv2.imshow('Face Detected',img)
    cv2.waitKey(1)
```

FOR THE INPUT VIA SCREEN CAPTURE

```
#Required Libaries
import cv2
import numpy as np
import face_recognition
import os
from PIL import ImageGrab
#Importing Data
path = 'Data'
images = []
classNames = []
myList = os.listdir(path)

for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)
```

```
def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList

def captureScreen(bbox=(300,300,690+300,530+300)):
    capScr = np.array(ImageGrab.grab(bbox))
    capScr = cv2.cvtColor(capScr, cv2.COLOR_RGB2BGR)
    return capScr

encodeListKnown = findEncodings(images)
print(len(encodeListKnown))
print('Encoding Complete')

while True:
    img = captureScreen()
    imgS = cv2.resize(img,(0,0),None,1,1)
    imgS = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    facesCurFrame = face_recognition.face_locations(imgS)
    encodesCurFrame = face_recognition.face_encodings(imgS,facesCurFrame)

    for encodeFace,faceLoc in zip(encodesCurFrame,facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown,encodeFace)
```

```
        faceDis = face_recognition.face_distance(encodeListKnown,encodeFace)
        #print(faceDis)
        matchIndex = np.argmin(faceDis)

        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            y1,x2,y2,x1 = faceLoc
            cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)
            cv2.rectangle(img,(x1, y2-30), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img,name,(x1+9,y2-9), cv2.FONT_HERSHEY_PLAIN,1, (255,255,255)
,2)

        else:
            name = 'Unknown'
            y1, x2, y2, x1 = faceLoc
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_PLAIN, 1, (255,
255, 255), 2)

    cv2.imshow('Face Detected',img)
    cv2.waitKey(1)
```

APPENDIX 2

IMPLEMENTATION SCREENSHOT

Image Enrollment

Figure A 1.1 Image Input

A face recognition project uses images or video frames of faces as input, which are captured by a camera or video device. The system analyzes the input and extracts features unique to each individual such as facial shape, texture and uses them to identify or verify the person. The output is a binary decision or a confidence score indicating the likelihood of a match.

Face Detection

Figure A 1.2 Faces Detected in the Input

The output of a face detector in a face recognition project is a set of coordinates, usually in the form of a bounding box, that indicate the location of the face in the input image or video frame. It also includes a confidence score indicating the likelihood that the detected region is a face. The output is used as input for the next step, which is face recognition.

Data Scrapping

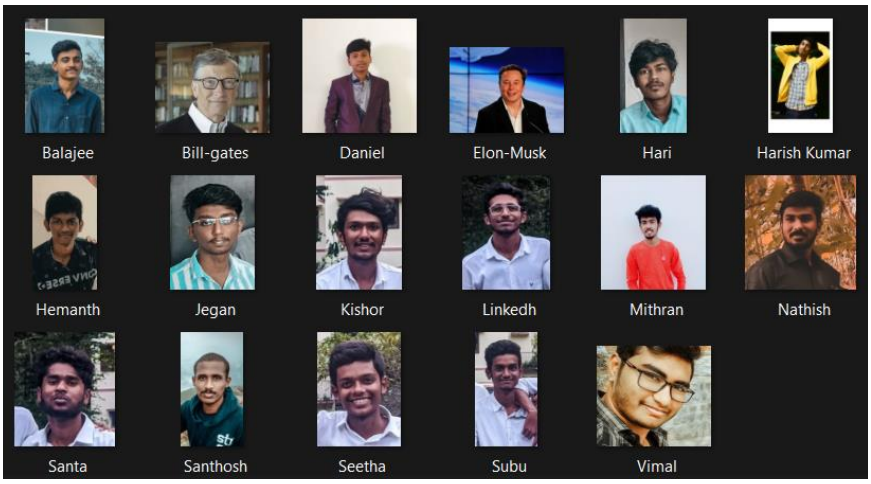

Figure A 1.3 Data Collected for the Encoding Faces

Data collection in a face recognition project involves gathering images or video frames of faces from different individuals, which will be used to train and test the system. The data should be diverse and representative of the population the system will be used with, including different ages, genders, ethnicities, poses, lighting conditions and expressions. The data should be labeled with the appropriate information for each individual, such as their name or ID.

Face Recognizer

Figure A 1.4 Face Recognition with Names

The output of a face recognizer in a face recognition project is a binary decision or a confidence score indicating the likelihood of a match between a given face and the faces in the database. It also includes the identification of the person matched, if any. The output can be used for different purposes such as access control, security or identification.