A
Mini Project Report on

FACE RECOGNITION USING IMAGE PROCESSING

Submitted To
Jawaharlal Nehru Technological University Hyderabad
In Partial Fulfillment of the Requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted By

K.V.HEMANTH	207Z1A1225
SHELDON SHAJI	207Z1A1241
V.NIKIL	207Z1A1248

Under the Guidance of
Mr. T. Madhu
Associate Professor

SCHOOL OF ENGINEERING
Department of Information Technology

NALLA NARASIMHA REDDY
EDUCATION SOCIETY'S GROUP OF INSTITUTIONS
(Approved by AICTE, New Delhi, Affiliated to JNTU-Hyderabad)
Chowdariguda (VIll) Korremula 'x' Roads, Via Narapally, Ghatkesar (Mandal)
Medchal (Dist), Telangana-500088

2023-2024

NALLA NARASIMHA REDDY
Education Society's Group of Institutions - Integrated Campus
Approved by AICTE, New Delhi, Affiliated to JNTU - Hyderabad
SCHOOL OF ENGINEERING
DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project report titled **"Face Recognition Using Image Processing"** is being submitted by **K.V.Hemanth (207Z1A1225), Sheldon Shaji (207Z1A1241)**, and **V.Nikil (207Z1A1248)** in Partial fulfillment for the award of **Bachelor of Technology in Information Technology** is a record bonafide work carried out by them. The results embodied in this report have not been submitted to any other University for the award of any degree.

Internal Guide
(Mr.T.Madhu)

Head of the Department
(Dr.K.Rameshwaraiah)

Submitted for the University Examination held on…….

External Examiner

DECLARATION

We **K.V.Hemanth, Sheldon Shaji and V.Nikil** are students of **Bachelor of Technology in Information Technology, Nalla Narasimha Reddy Education Society's Group of Institutions**, Hyderabad, Telangana State, hereby declare that the work presented in this project work entitled **Face Recognition Using Image Processing** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of engineering ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning.

K.V.Hemanth	207Z1A1225
Sheldon Shaji	207Z1A1241
V.Nikil	207Z1A1248

Date:

Signature:

ACKNOWLEDGEMENT

We express our sincere gratitude to our guide **Mr.T.Madhu,** Associate Professor, NNRESGI, who motivated throughout the period of the project and also for his valuable and intellectual suggestions, guidance, and constant encouragement right throughout our work.

We would like to express our profound gratitude to our mini project in-charge **Mrs.Ch.Ramya,** Assistant Professor**,** NNRESGI, for her support and guidance in completing our project and for giving us this opportunity to present the project work.

We profoundly express thanks to **Dr.K.Rameshwaraiah,** Professor & Head, Department of Information Technology, NNRESGI, for his cooperation and encouragement in completing the project successfully.

We wish to express our sincere thanks to **Dr.G.Janardhana Raju,** Dean School of Engineering, NNRESGI, for providing the facilities for completion of the project.

We wish to express our sincere thanks to **Dr**.**C.V.Krishna Reddy,** Director, NNRESGI, for providing the facilities for completion of the project.

Finally, we would like to thank overall mini-project coordinator, members of Project Review Committee (PRC), all the faculty members and supporting staff, Department of Information Technology, NNRESGI, for extending their help in all circumstances.

By

K.V.Hemanth	207Z1A1225
Sheldon Shaji	207Z1A1241
V.Nikil	207Z1A1248

ABSTRACT

Face recognition from picture or motion picture could be a standard point in bio science analysis. The field of computer vision has experienced remarkable growth in recent years, finding widespread applications in areas as diverse as security, human-computer interaction, and surveillance. An important aspect of face recognition is preprocessing, where images are subsequently cleaned and normalized to improve the accuracy of word recognition. These papers explore various pre-processing techniques such as image resizing, illumination stabilization, and angle filtering. The main objective of this project is to recognize the face. The system subsequently to keep track on the image for further details to identify a face in the image. It serves as an individual identity of everyone and therefore face recognition helps in authenticating any person's identity using his personal characteristics. It consists of static image where it detects the face and details where it apprehends the occasions that manifest the image. The most essential goal of project is to recognize the face and provide the details for an image.

Keywords: Face Recognition, NumPy, OpenCV, CNN, Image Processing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S.NO.	ABBREVIATION	DEFINITION
1	OpenCV	Open-source Computer Vision
2	UML	Unified Modeling Language
3	ANN	Artificial Neural Networks
4	CNN	Convolutional Neural Network
5	DFD	Data Flow Diagram

1. INTRODUCTION

Face identification is an important task in many settings, such as police stations and in the events. Traditionally, Image in the police stations or in any events has been recorded manually using methods such as applications sheets or punch sheets. To address these challenges, we have developed a face recognition using Image processing system. This system uses advanced computer vision techniques to detect and recognize faces in an image, and then uses this information to give the detail of the image. The goal of this documentation is to provide an overview of the Face recognition using Image Processing, including its features, capability, and user interface. We hope that this documentation will help users understand how the system works and how to effectively use it for attendance tracking.

1.1 MOTIVATION

In today's networked world the need to maintain security of information or physical property is becoming both increasingly important and increasingly difficult. No automatic systems are there that can track person's activity. If we will be able to track Facial recognition of persons automatically then we can find the criminal easily since facial images changes. So, we decided to make a Facial Recognition System. We are interested in this project after we went through few papers in this area. The papers were published as per their system creation and way of creating the system for accurate and reliable facial recognition system. As a result, we are highly motivated to develop a system that recognize faces and track one person's activity.

1.2 PROJECT STATEMENT

The Images tracking methods, such as applications such as FIRs in the police stations and also in hospitals are often time-consuming and prone errors. In addition, these applications cannot be for long amount time, and also it is time-consuming Process. To address these challenges, we have developed a face recognition using Image processing system. This system uses advanced computer vision techniques to detect and recognize faces in an image, and then uses this information to give the detail of the image. The system will provide a quick and convenient way to track faces in the image, while also reduce the risk of errors and improving the system efficiency.

1.3 OBJECTIVE OF PROJECT

The objective of our project is to develop a face recognition system. To experiment machine learning algorithm in Image processing. To detect the face in the image and provide the details of that Image. This process will no take more amount of time. It uses some machine learning algorithms to identify and recognize the faces and provide the detail of the faces in the image.

1.4 LIMITATIONS OF PROJECT

Privacy concerns:

Some Users may be concerned about their privacy when using a face recognition for images. It may be necessary to address these concerns and ensure that the system is used in a way that respects user's privacy.

Maintenance and updates:

The system may require regular maintenance and updates to ensure optimal performance. This may involve additional time and resources.

Dependence on technology:

The system may depend on technology such as cameras, images and servers, which could potentially fail or experience downtime. This could disrupt image Tracking or recognizing.

2. LITERATURE SURVEY

2.1 INTRODUCTION

1. Face Detection and Recognition System using Digital Image Processing

Authors: Gurlove Singh and Amit Kumar Goel

[1] It serves as an individual identity of everyone and therefore face recognition help authenticating any person's identity using his personal characteristics. The whole procedure for authenticating any face data is sub-divided into two phases, in the first phase, the face detection is done quickly except for those cases in which the object is placed quite far, followed by this the second phase is initiated in which the face is recognized as an individual. Basically, there are two type of techniques that are currently being followed in face recognition pattern that is, the Eigenface method and the Fisher face method. The Eigenface method basically make use of the PCA (Principal Component Analysis) to minimize the face dimensional space of the facial features. The area of concern of this paper is using the digital image processing to develop a face recognition system. Since the no. of Eigen faces to be used is restricted in PCA transformation that's why the system did not have an accuracy of more than 90% for both manual and automatic face recognition. The performance of this system can be compared with the manual face detection only if we integrate the eye detection system with the developed system.

2. Face Recognition in Surveillance System

Authors: loshila Grace.L. K and K. Reshmi

[2] Face identification has made its presence evident as an impact of advancement in technologies. Face identification in video surveillance has always been an challenging field where it needs to undergo a series of consideration to identify a particular face in the video In this paper initially a real time facial detection is performed using an open source environment that runs in processing and later on captured faces are correlated with the template faces that are stored in the database using MATLAB .once the faces are identified then the profile of the person is displayed to notify the behavioral status of the person who is under study. face identification has always been research study in spite of other biometric

techniques such as iris recognition, fingerprint etc. since human faces are dynamic in nature which possess higher degree of variability in its size and shape that sometime produce faulty results as computers make misconception in identifying them.

3. Face Recognition Algorithm Based on Image Processing

Authors: Yan Sun, Zhengyuan Ren, and Wenxi Zheng

[3] The most important thing for everyone is information security. In order to improve the security level of network information and identify and detect faces, the method used in this paper has improved compared with the traditional AdaBoost method and skin color method. AdaBoost detection is performed on the image, which reduces the probability of false detection. The experiment compares the experimental results of the AdaBoost method, the skin color method and the skin color+ AdaBoost method. All operations in the KPCA and KFDA algorithms are performed by the inner product kernel function defined in the original space, and no specific non-linear mapping function is involved. The full name of KPCA is kernel principal component analysis. The full name of KFDA is kernel Fisher discriminant analysis. Combining the zero- space method kernel discriminant analysis method improves the ability of discriminant analysis to extract non-linear features. Through the secondary extraction of PCA features, a better recognition result than the PCA method is obtained. This paper also proposes a zero-space based Fisher discriminant analysis method. Experiments show that the zero-space-based method makes full use of the useful discriminant information in the zero space of the intraclass dispersion matrix, which improves the accuracy of face recognition to some extent. If you choose the polynomial kernel function, when d=0.8, KPCA has a higher recognition ability. When d=2, the recognition rate of KFDA and zero space-based KFDA is the largest. For polynomial functions, in general, d=2.

2.2 EXISTING SYSTEM

In the existing system, classification is done through eigenfaces method. The work includes the application of face recognition. The eigen faces to represent the feature vector for human faces. It uses eigen vectors and PCA to recognize the faces. It uses some statistical methods to recognize the faces in the images.

Disadvantages:

- It recognizes only one face at each time.
- Time consuming.

2.3 PROPOSED SYSTEM

A system is designed that can recognize and track human faces and give necessary information as per the training data. It trains the set of information then it finds in line with them no knowledge loss happens. Therefore, we can simply acknowledge the feel and classification done by Neural Networks. In addition to that, recognize the image and provide the name for the image in the rectangular box. It uses CNN Algorithm.

Advantages:

- It recognizes multiple faces in an image at once.
- Fast and easy implementation.

3. SYSTEM ANALYSIS

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

3.1 INTRODUCTION

In this phase the requirements are gathered and analyzed. User's requirements are gathered in this phase. This phase is the main focus of the users and their interaction with the system.

- What is being done?
- How is it being done?
- Who is doing it?
- When is he doing it?
- Why is it being done?
- How can it be improved?

These general questions are answered during a requirement gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.
Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

3.2 SOFTWARE REQUIREMENTS

It deals with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed. The software requirements are

description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view. We should try to understand what sort of requirements may arise in the requirement elicitation phase and what kinds of requirements are expected from the software system.

A software requirement can be of 3 types:

- Functional requirements
- Non-functional requirements
- Domain requirements

3.2.1. Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. For example, in a hospital management system, a doctor should be able to retrieve the information of his patients. Each high-level functional requirement may involve several interactions or dialogues between the system and the outside world. In order to accurately describe the functional requirements, all scenarios must be enumerated. There are many ways of expressing functional requirements e.g., natural language, a structured or formatted language with no rigorous syntax and formal specification language with proper syntax.

- Data Collection
- Data Pre-processing
- Training and Testing

3.2.2. Non-Functional Requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements. They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance

3.2.3. Domain Requirements

Domain requirements are the requirements which are characteristic of a particular category or domain of projects. The basic functions that a system of a specific domain must necessarily exhibit come under this category. For instance, in an academic software that maintains records of a school or college, the functionality of being able to access the list of faculty and list of students of each grade is a domain requirement. These requirements are therefore identified from that domain model and are not user.

The software requirements that are required for this project are as follows:

- Operating System : Windows XP and Above
- Technology : Image Processing
- Coding Language : Python
- UML'S : Star UML
- Editor : Jupiter Notebook

3.3. HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, a hardware requirements list is often accompanied by a hardware compatibility list, especially in case of operating systems. The hardware requirements that are required for this project are as follows:

- Processor : Dual Core and Above
- RAM : 4 GB
- Hard Disk : 50GB and Above

3.4. CONTENT DIAGRAM

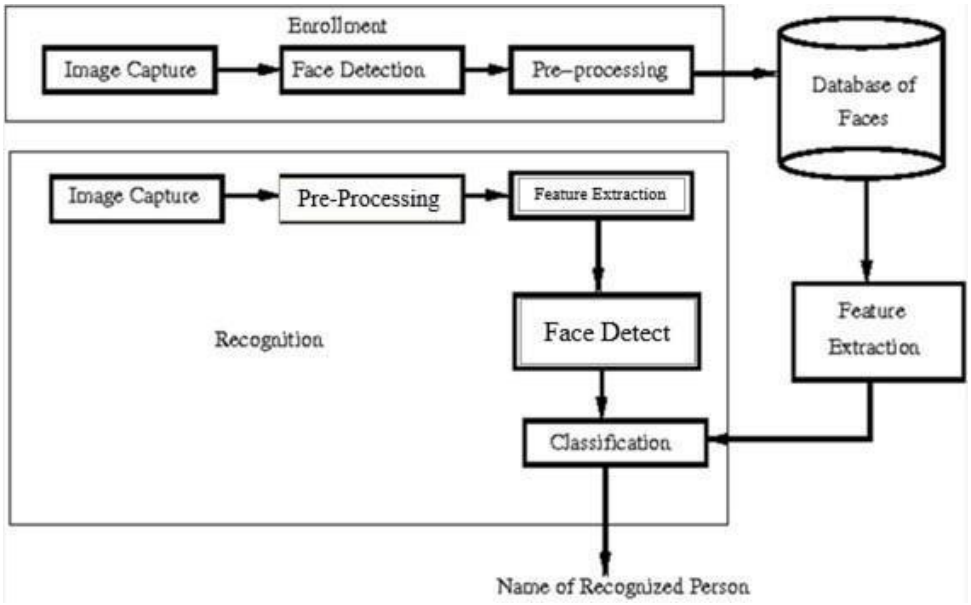


Fig: 3.4.1 Block diagram of System Architecture

The above diagram describes about the architecture of the face recognition system. It performs some of the operations and provide the result of the details of the image. The operations named as Image capture, Pre-processing, Feature Extraction, Face detect and Classification and also it performs verification. Later, it provides the output as name of the recognized person.

4. SYSTEM DESIGN

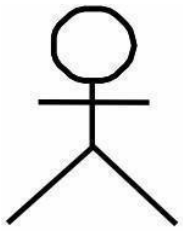
4.1. INTRODUCTION

Software design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints. Software design may refer to either "all the activity involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems" or "the activity following requirements specification and before programming, as in a stylized software engineering process." Software design usually involves problem solving and planning a software solution. This includes both a low-level component design and a high-level, architecture design. Design is the first step in the development phase for anytechniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves four technical activities – design, coding, implementation and testing that are required to build and verify the software.

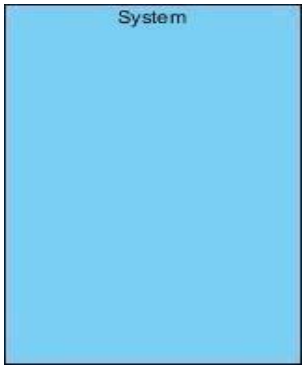
The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Actor: An actor is a model element that interacts with a system.



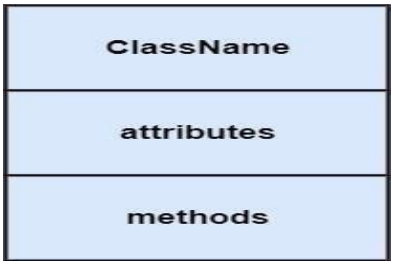
Use Case: A use case describes a function that a system performs to achieve the user's goal.

Use case system: A system of a use case defines and represents boundaries of a business, software system, physical system or device, subsystem, component or even single classin relation to the requirements gathering and analysis.



Associations: A line between actors and use cases. It describes which actors areassociated with which use cases.

Class: A class represents an object or a set of objects that share a common structure and behavior.

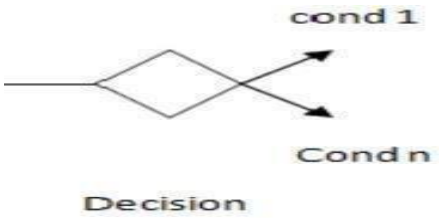


Generalization: A generalization is a relationship between a parent class and a child class. In this, the child class is inherited from the parent class.

Composition: It portrays the dependency between the parent and its child.

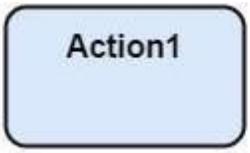
Control flow: The control flow determines the flow within an activity.

Decision Box: It makes sure that the control flow or object flow will follow only one path.

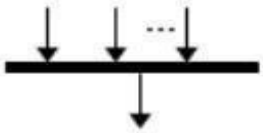


Initial State: It depicts the initial stage or beginning of the set of actions.

Action: An action is a named element that is the fundamental unit of an executable functionality.

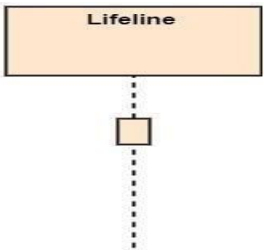


Join: Join nodes are used to support concurrent activities converging into one.

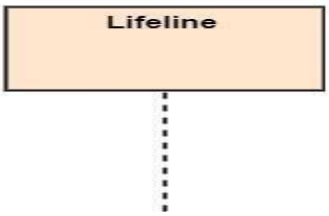


Call Message: It depicts workflow or activity over time using messages passed from element to element.

Activation: It is represented by a thin rectangle on the life line. It describes that time period in which an operation is performed by an element.



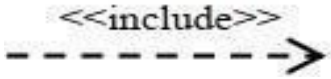
Life Line: Life Line represents the objects that participates in an interaction.



Final State: It is the stage where all the control flows and object flows end.

Final State

<<Include>>:-<<Include>> relationship represents behavior that is factored out of the use case.



4.2. UML DIAGRAMS

UML stands for Unified Modelling Language which is used in object-oriented software engineering. It is a standard language for specifying, visualizing, constructing, and documenting the artefacts of the software systems. UML is different from other common programming languages like C++, Java, and COBOL etc. It is pictorial language used to make software blueprints.

Although typically used in software engineering it is a rich language that can be used to model an application structure, behavior and even business processes. There are 8 UML diagram types to help us model this behavior.

Characteristics of UML

- The UML has the following features:
- It is a generalized modelling language.
- It is distinct from other programming languages like C++, Python, etc.
- It is interrelated to object-oriented analysis and design.
- It is used to visualize the workflow of the system.
- It is a pictorial language, used to generate powerful modelling artifacts.

UML is linked with object-oriented design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:

There are two types of UML modelling:

- Structural Modelling
- Behavioral Modelling

Structural Modelling

Structural model represents the framework for the system and this framework is the place where all other components exist. Hence, the class diagram, component diagram and deployment diagrams are part of structural modelling. They all represent the elements and the mechanism to assemble them.

Structural Modelling captures the static features of a system. They consist of the following:

- Classes diagrams
- Objects diagrams
- Deployment diagrams
- Package diagrams
- Composite structure diagram
- Component diagram

Behavioral Model

Behavioral model describes the interaction in the system. It represents the interaction among the structural diagrams. Behavioral modelling shows the dynamic nature of the system.

They consist of the following:

- Activity diagrams
- Interaction diagrams
- Use case diagrams

All the above show the dynamic sequence of flow in a system.

4.3. DATA FLOW DIAGRAM

A data flow diagram (DFD) is a visual representation of the information flow through a process or system. DFDs help you better understand process or system operations to discover potential problems, improve efficiency, and develop better processes. They range from simple overviews to complex, granular displays of a process or system. There are two types of DFDs — logical and physical. Logical diagrams display the theoretical process of moving information through a system, like where the data comes from, where it goes, how it changes, and where it ends up. Physical diagrams show you the practical process of moving information through a system. It can show how your system's specific software, hardware, files, employees, and customers influence the flow of information.

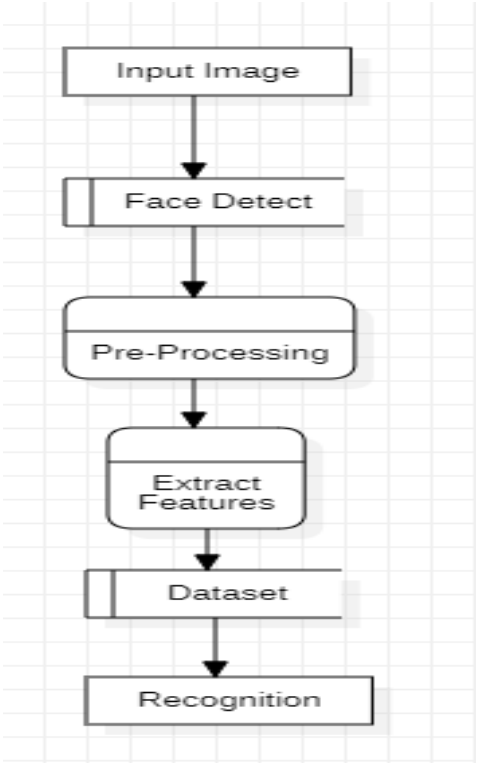


Fig 4.3: DFD of face recognition.

The above Data Flow Diagram show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. It consists of Data flow and also the processes and data store.

4.4. USE CASE DIAGRAM

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. The "actors" are people or entities operating under defined roles within the system.

As the most known diagram type of the behavioral UML diagrams, use-case diagrams give a graphic overview of the characters involved in a system, different functions needed by those characters and how these different functions are interacted.

Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.

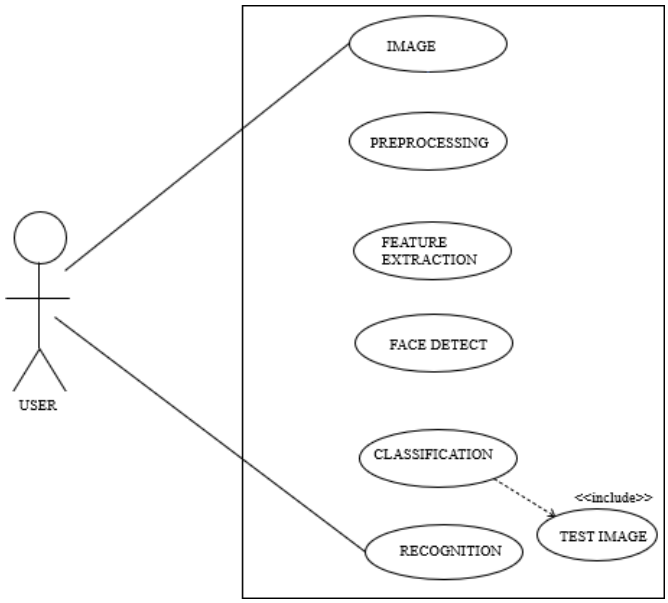


Fig 4.4: Use Case diagram of face recognition.

The above use case diagram represents the interaction only with user, how the user works and its behavior where common people can understand in easy manner. Face recognition shows the faces identified or performed by the user and also actions are also performed by the user and each use case represents about the function what the user can perform to achieve the goal.

4.5. CLASS DIAGRAM

Class diagrams are the main building blocks of every object-oriented method. It is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

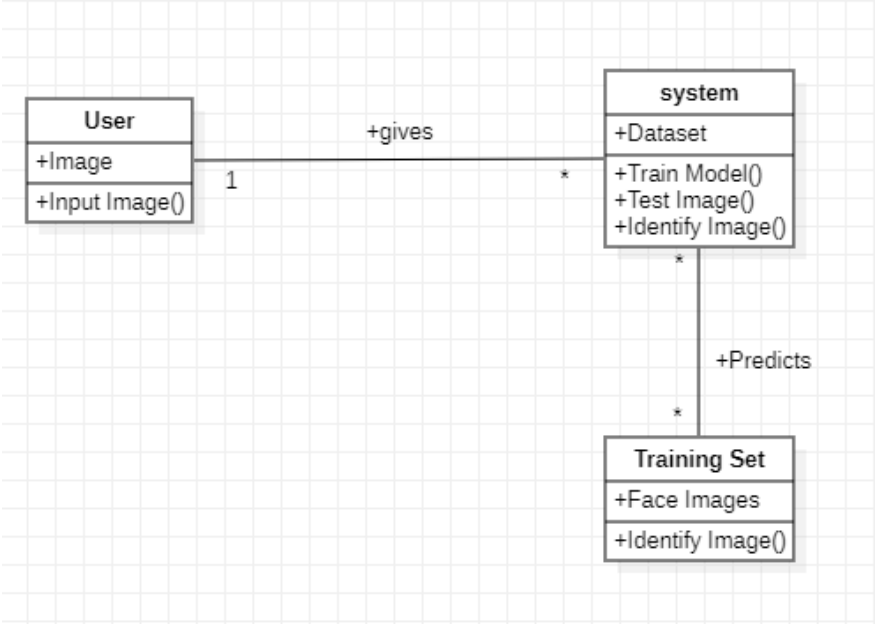


Fig 4.5: Class Diagram of face recognition

The above class diagram describes the attributes and operations done in face recognition system. In class diagram, class represents about the objects where in face recognition system there are about three classes present system, user, Training Set.

4.6. SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence diagrams emphasize on time sequence of messages and are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

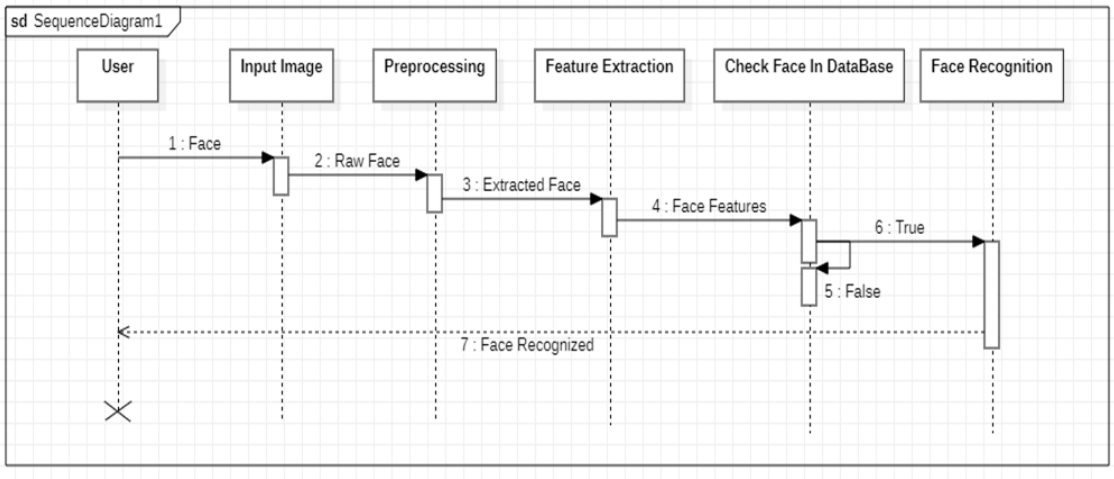


Fig 4.6: The Sequence Diagram of face recognition

The Sequence diagram represents the flow of messages performed in the face recognition system. Sequence diagram illustrates the communication between the user to achieve the face recognition. When the user uploads the image into the system, and the system performs all the operations and provide the output to the user.

4.7. ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

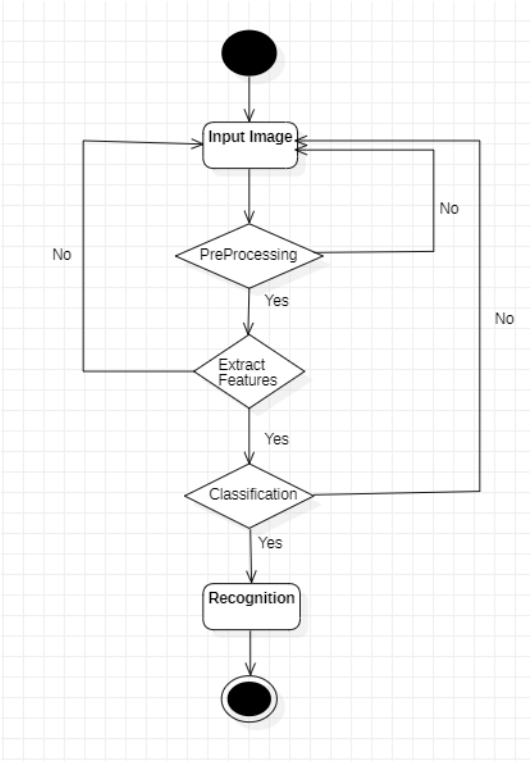


Fig 4.7: The Activity Diagram of face recognition.

The Activity Diagram represents the flow of activities done by the system. It consists decisions symbols which represents both the negative and the positive cases. In Face recognition system if the flow goes in invalid cases our system displays to correct the actions and each node in the system represents how the face recognition process goes on.

5. IMPLEMENTATION AND RESULTS

5.1. INTRODUCTION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system is giving confidence on the new system for the users that it will work efficiently and effectively. Functions is an essential way of developing a software. Functions are used for writing the code which is to be repeated several times. For example, if we need same code, then we must have to write that code again and again. So, in order to remove this, we use functions.

The system can be implemented only after thorough requirement analysis and designing of the proposed system. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over an evaluation of change over methods apart from planning.

5.2. METHOD OF IMPLEMENTATION

- **Image Capture**

Here camera captures the image of the user and take as input using Image, Image is the major thing in the project we can't even run the project without Image. A facial recognition system uses biometrics to map facial features from a photograph. It compares the information with a database of known faces to find a match.

There is no need of fixed Position of the face it means we may give a still as our comfort zone. We Can wear spectacles also by wearing spectacles we can extract the features using the nose, mouth and eyebrows etc.,

- **Color Image Processing:**

A computer sees an image as 0s and 1s. Pixel is the smallest unit in an image. When we take a digital image, it is stored as a combination of pixels. Each pixel contains a different If it a grayscale image, it has only one pixel, whereas if it is a colored image, it contains three channels: red, green and blue.

A digital image represented as pixels and channels. A digital-colored image, each channel of each pixel has a value between 0 and 255.

Convert color to greyscale: (A grayscale image is a monochrome that uses the tonal range of gray)

Import the OpenCV and read the original image using imread() than convert to grayscale using cv2. cvtcolor() function. Grayscale simplifies the algorithm and reduces computational requirements.

Convert greyscale to black and white:(A grayscale image is a monochrome that uses the tonal range of gray).

Converting an image to black and white with OpenCV can be done with a simple binary thresholding operation for each pixel of the gray scale image, if its value is lesser than the threshold, then we assign to it the value 0 (black). Otherwise, we assign to it the value 255 (white). This transformation is useful in detecting blobs and further reduces the computational complexity.

- **Neural Network**

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity.

An Artificial Neural Network is made up of 3 components: Input Layer, Hidden (computation) Layers, Output Layer. We can add number of hidden layers. A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. Neural networks can help computers make intelligent decisions with limited human assistance.

- **Haar cascade**

The feature is the pixel value in the write subtracted from the pixels value in the blank area. Haar cascade classifier is trained in given some input faces and non-faces and training a classifier that identifies a face. Haar cascade is an algorithm that can detect objects in images, irrespective of their scale in image and location. Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

- **Positive images**– These images contain the images which we want our classifier to identify.
- **Negative images**– Images of everything else, which do not contain the object we want to detect.

5.3. EXPLANATION OF KEY FUNCTIONS

NumPy: NumPy is a powerful tool for working with numerical data, and is particularly useful for data manipulation, statistical analysis, and machine learning.

OpenCV: OpenCV (Open-source Computer Vision) is a library of computer vision and machine learning algorithms, designated to help developers build applications that can process and analyze visual data. It is used for image processing, video analysis, computer version etc.

Operating System: The ‘os’ module is a built-in Python library that provides functions for interacting with the operating system. It allows you to perform a variety of tasks, such as reading and writing files, accessing and manipulating directories, and running system commands.

Face Recognition: Facial recognition can identify human faces in images or videos, determine if the face in two images belongs to the same person, or search for a face among a large collection of existing images.

Tkinter: Tkinter reacts to user input, changes from your program, and even refreshes the display only when actively running an event loop. If your program isn't running the event loop, your user interface won't update.

File Dialog: File dialog is one of the tkinter modules that provides classes and library functions to create file/directory selection windows. You can use file dialog where you need to ask the user to browse a file or a directory from the system.

Source Code

```
import tkinter as tk
from tkinter import filedialog
import face_recognition as fr
import cv2
import numpy as np
import os

def browse_file():
    global test_image_path
    test_image_path = filedialog.askopenfilename(filetypes=[("Image files", "*.jpg *.jpeg *.png *.bmp")])
    test_image_entry.delete(0, tk.END)
    test_image_entry.insert(0, test_image_path)

def recognize_faces():
    path = "./train/"
    known_names = []
    known_name_encodings = []

    images = os.listdir(path)
    for _ in images:
        image = fr.load_image_file(path + _)
        image_path = path + _
        encoding = fr.face_encodings(image)[0]

        known_name_encodings.append(encoding)
        known_names.append(os.path.splitext(os.path.basename(image_path))[0].capitalize())

    test_image = test_image_path
    image = cv2.imread(test_image)

    face_locations = fr.face_locations(image)
    face_encodings = fr.face_encodings(image, face_locations)
    for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
```

```
        matches = fr.compare_faces(known_name_encodings, face_encoding)
        name = "Unknown User"  # Default label for unknown faces

        face_distances = fr.face_distance(known_name_encodings, face_encoding)
        best_match = np.argmin(face_distances)

        if matches[best_match] and face_distances[best_match] < 0.5:
            name = known_names[best_match]
        cv2.rectangle(image, (left, top), (right, bottom), (0, 0, 255), 2)
        cv2.rectangle(image, (left, bottom - 15), (right, bottom), (0, 0, 255), cv2.FILLED)
        font = cv2.FONT_HERSHEY_DUPLEX
        cv2.putText(image, name, (left + 6, bottom - 6), font, 0.35, (255, 255, 255), 1)
    cv2.imshow("Result", image)
    cv2.imwrite("./output.jpg", image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
# Create the main window
root = tk.Tk()
root.title("Face Recognition")
root.geometry("400x300")
# Test image selection
test_image_label = tk.Label(root, text="Select Test Image:")
test_image_label.pack()
test_image_entry = tk.Entry(root)
test_image_entry.pack()
browse_button = tk.Button(root, text="Browse", command=browse_file)
browse_button.pack()
browse_button.place(x=160,y=70)
# Recognize faces button
recognize_button = tk.Button(root, text="Recognize Faces", command=recognize_faces)
recognize_button.pack()
recognize_button.place(x=150,y=120)

root.mainloop()
```

5.4. OUTPUT SCREENS

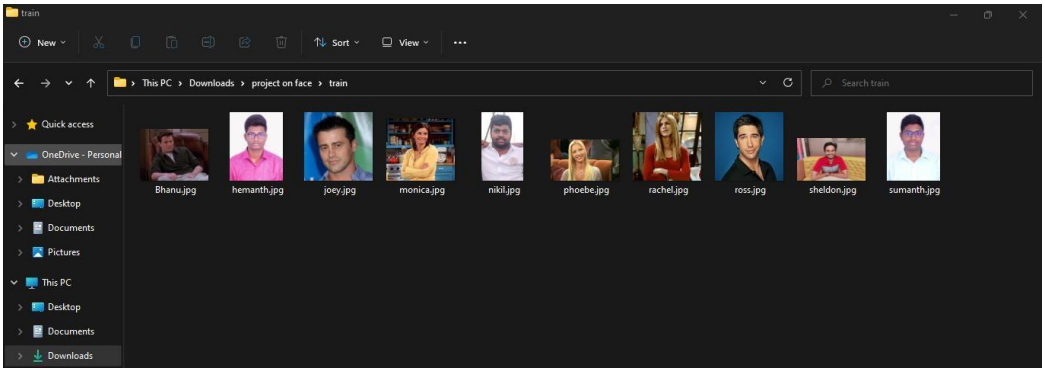


Fig 5.4.1: Training Set

These are the training Images for the Face Recognition Model where we have provided few of the images with names when the face will recognized in the test then the names for the image will be displayed.

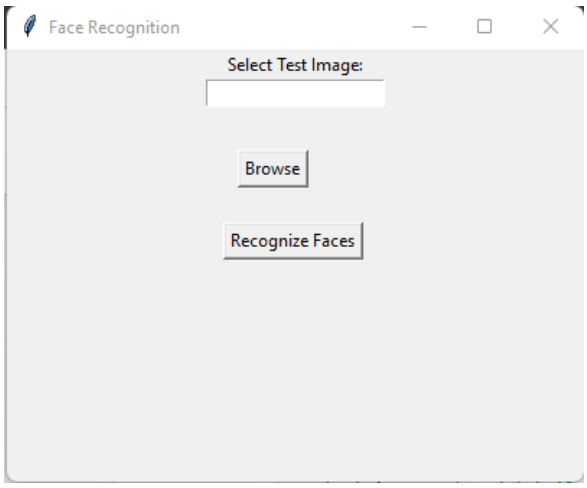


Fig 5.4.2: Browse Button

The Browse Button is used to select the images from the Test Folder in the System. It will be used to find the Images in the system and the browse button will only accepts the images in the format of the jpeg, jpg, png and other images extensions.

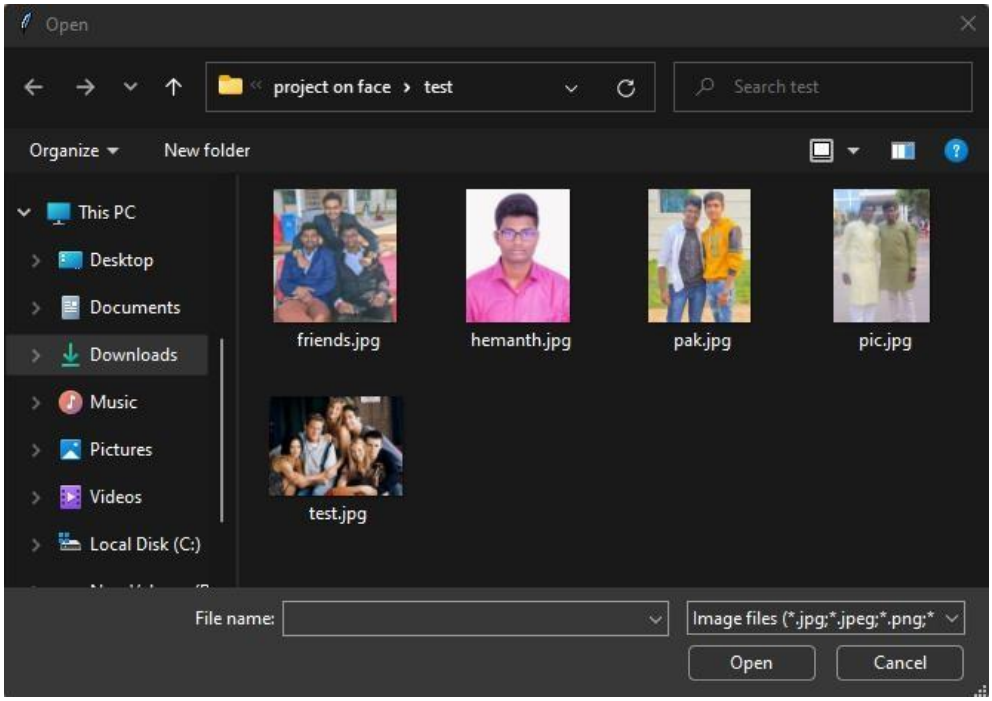


Fig 5.4.3: Select Test Image

By clicking the Browse button, it will open the folder where we will be able to get the above image. Then select any one of the lists of images in the test folder and then “click ok” to fetch that image to test.

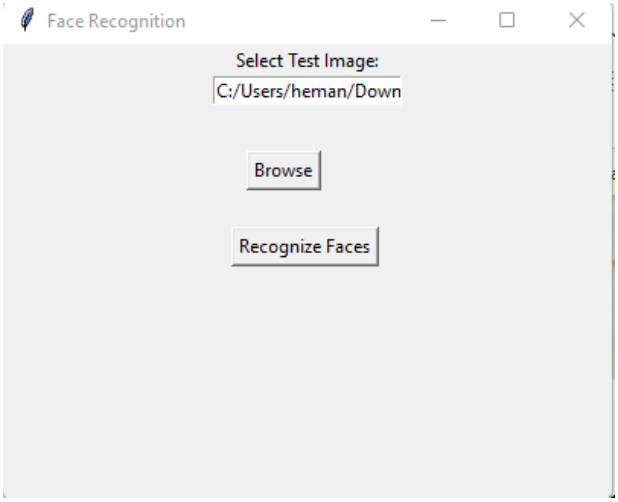


Fig 5.4.4: Recognize

After Clicking “Ok” we will be able to get the path of that image in the text field and then we will click the Recognize faces to get the detail of the image from the trained set.

Fig 5.4.5: output of all faces recognized

This above image describes about all the face of the image has been recognized according to the Training dataset. The faces in the image have bordered with the red color and also the name has font of the Hershey duplex. Also, we can increase the font of the name as we required.

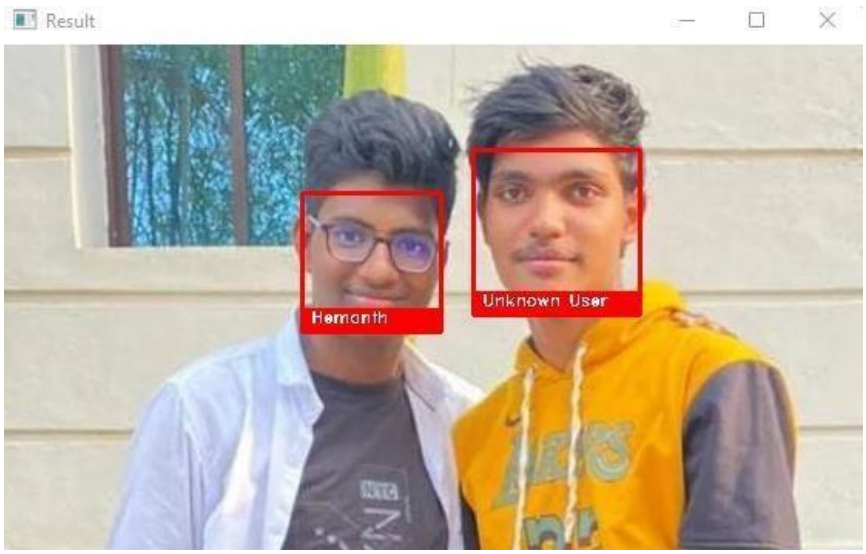


Fig 5.4.6: output of face of Unknown User

This above image describes about all the face of the image has been recognized according to the Training dataset and also if the image is not in the training dataset, then it provides the name of the face as the unknown user. The faces in the image have bordered with the red color and also the name has font of the Hershey duplex. Also, we can increase the font of the name as we required.

6. TESTING

6.1. INTRODUCTION

Software testing takes place during software engineering. It is done before the release to the final audience.

Software testing is meant to see how the software works under different conditions. These conditions might be different depending on what the audience is. Testing is done to understand if it will work correctly, partially fail to work properly, or totally fail to work properly. Each test may be used to see how one, or many, parts of the software work at a point in its development. A review of the results of tests may show that some parts of the software system may need to be done again, or may work well. Some bad performances or software bugs may need to be fixed. After more work on the software, testing may be done again.

For larger software systems, tracking may take place checking completeness of the set of tests, test results, and how quickly any problems are fixed. All this information can be used for decision making about how ready the software is, and when it could be released to the final audience. Software testing may be done with separate parts of the software, with a group of these parts, or with the entire software. Software testing may be done by allowing the software to be used by a small number of people who the software is meant for, under controlled settings.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

Software testing can be divided into two steps:

Verification: it refers to the set of tasks that ensure that software correctly implements a specific function.

Validation: it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Verification: "Are we building the product, right?"

Importance of Software Testing:

To put it simply, the importance of software testing can be traced from the user's response. It assures the quality of the product and satisfies the customers as well as users. Also, it promises better business optimization (less maintenance cost), reliability, and superior user experience. Substantially, the iterative efforts spending to mound a powerful yet error-free software are far and wide. Despite the designers and developers, software testing plays a decisive role in the process. As the development process follows the software development lifecycle, the testing process lines up right behind the queue to compile the units to verify and validates.

Altogether, it is a continuous process of verification and validation to deliver an excellent product. The importance of software testing is evident in the final quality assurance report. By successfully passing through the diversified levels of testing, the final products could perform beyond the expected outcomes. At each level, the tester scan not only figure out the errors but also prevent such snags in the future. Besides, exploring each mistake makes for the birth of an improvised version of the software.

Types of Software Testing:

Manual Testing:

Manual testing includes testing software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

- Testers use test plans, test cases, or test scenarios to test software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.
- Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software. The difference between expected output

and output, given by the software, is defined as a defect. The developer fixed the defects and handed it to the tester for retesting.

- Manual testing is essential because one of the software testing fundamentals is "100% automation is not possible".

There are various methods used for manual testing. Each technique is used according to its testing criteria. Types of manual testing are given below:

- White Box Testing
- Black Box Testing
- Gray Box Testing

Goal of Manual Testing:

The key concept of manual testing is to ensure that the application is error free and it is working in conformance to the specified functional requirements. Test Suites or cases, are designed during the testing phase and should have 100% test coverage. It also makes sure that reported defects are fixed by developers and re-testing has been performed by testers on the fixed defects. Basically, this testing checks the quality of the system and delivers bug free product to the customer.

Advantages of Manual Testing:

- It does not require programming knowledge while using the Black box method.
- It is used to test dynamically changing GUI designs.
- Tester interacts with software as a real user so that they are able to discover usability and user interface issues.
- It ensures that the software is a hundred percent bug-free.it is cost-effective.
- Easy to learn for new testers.

Disadvantages of Manual Testing:

- It requires a large number of human resources.
- It is very time-consuming.

Manual testing tools:

In manual testing, different types of testing like unit, integration, security, performance, and bug tracking, we have various tools such as Jira, Bugzilla, Mantis, Zap, Tessy, Load Runner, Citrus, SonarQube, etc. available in the market. Some of the tools are open-source, and some are commercial.

Automated Testing:

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves the automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly. Apart from regression testing, automation testing is also used to test the application from a load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing.

Software Testing Automation Process:

The software testing automation process in a reputed firm is an integral part in the process of manufacturing any new software product. In view of the fact that technological equipment has many uses in our lives; we are constantly looking for newer as well as more advanced technological tools that can increase our rate of productivity as well as our efficiency as well. As a result of this, we find that there are a number of companies that are now manufacturing various software products. The software testing process assists the manufacturers in ensuring that there are no errors in the new software product that is being manufactured.

Automation Testing Process:

The software testing automation process in a reputed firm is an integral part in the process of manufacturing any new software product. In view of the fact that technological equipment has many uses in our lives; we are constantly looking for newer as well as more advanced

technological tools that can increase our rate of productivity as well as our efficiency as well.

As a result of this, we find that there are a number of companies that are now manufacturing various software products. The software testing process assists the manufacturers in ensuring that there are no errors in the new software product that is being manufactured.

6.2. TESTING ACTIVITIES

Software level testing can be majorly classified into 3 levels:

- Unit Testing
- Integration Testing
- System Testing

Unit Testing:

Unit Testing is a software testing technique by means of which individual units of software i.e., group of computer program modules, usage procedures and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which all independent modules are tested to determine if there are any issue by the developer himself. It is correlated with functional correctness of the independent modules. Unit Testing is defined as a type of software testing where in dividual components of a software are tested. Unit Testing of software product is carried out during the development of an application. An individual component may be either an individual function or a procedure. Unit Testing is typically performed by the developer. In SDLC, Unit testing is first level of testing done before integration testing. Unit testing is such type of testing technique that is usually performed by the developers. Although due to reluctance of developers to tests, quality assurance engineers also do unit testing.

Objectives of Unit Testing:

- To isolate a section of code.
- To verify the correctness of code.
- To test every function and procedure.
- To fix bug early in development cycle and to save costs.
- To help the developers to understand the code base & enable them to make.

Unit Testing Techniques:

The Unit Testing Techniques are mainly categorized into three parts which are Black box testing that involves testing of user interface along with input and output, White box testing that involves testing the functional behavior of the software application and Gray box testing that is used to execute test suites, test methods, test cases and performing risk analysis. Code coverage techniques used in Unit Testing are listed below:

- Statement Coverage
- Decision Coverage
- Branch Coverage
- Condition Coverage
- Finite State Machine Coverage

Unit Testing Advantages:

- Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit API.
- Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly (i.e., Regression testing).
- Due to the modular nature of the unit testing, we can test parts of the project without waiting for others to be completed.

Unit Testing Disadvantages:

- Unit testing can't be expected to catch every error in a program. It is not possible to evaluate all execution paths even in the most trivial programs.
- Unit testing by its very nature focuses on a unit of code. Hence it can't catch integration errors or broad system level errors.

Integration testing

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects for the time of interaction between integrated components or units. Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

Objectives of Integration Testing:

Integration testing reduces the risk of finding the defects in integrated components in the System testing phase. Integration defects can be complex to fix and they can be time consuming as well. Finding them early in the cycle eliminates the risk of making too many changes at the System testing phase. As each of the integrating components has been tested in the integration phase, the System testing can focus on end-to-end journeys and user specific flows.

- Reducing risk by testing integrating components as they become available. Verify whether the functional and non-functional behaviors of the interfaces are designed asper the specification
- To build confidence in the quality of the interfaces
- To find defects in the components, system or in the interfaces.
- Prevents defects from escaping to higher test levels of testing i.e., System testing.

Advantages of Integration testing:

- Integration testing provides a systematic technique for assembling a software system while conducting tests to uncover errors associated with interfacing.
- The application is tested in order to verify that it meets the standards set by the client as well as reassuring the development team that assumptions which were made during unit testing are correct.
- Integration testing need not wait until all the modules of a system are coded and unit tested. Instead, it can begin as soon as the relevant modules are available.

System Testing

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input.

The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or for the context of both.

System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS)System Testing is basically performed by a testing team that is independent for the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing.

Objectives of System Testing:

The goal of system testing is to minimize the risks associated with the behavior of the systemin a particular environment. For this, testers use the environment as close as possible to the one where a product will be installed after the release. The objectives are some smaller steps that allow achieving the goal. In system testing, there are several milestones that make the release of a flawlessly functioning system (read: software without critical bugs on production) possible. So, the primary objectives are:

- Reducing risks, for bug-free components don't always perform well as a system.
- Preventing as many defects and critical bugs as possible by careful examination.
- Verifying the conformance of design, features, and performance with the specifications stated in the product requirements.
- Validating the confidence in the system as a whole before moving to the final stage acceptance testing that takes place right before users get access to a product.

6.3. TEST CASES

Table 6.3.1: Test cases for face recognition

TEST CASE ID	TEST CASE DESCRIPTION	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	Train the Data	Import the Images as we required.	Need to Train the data.	Reading data is done.	Success
2	Detect facial features	Face bounding box region	Need to extract eye, mouth, nose contours	Feature extraction is done.	Success
3	Browse Button	We need to provide the Test Image	Select the Image and click Enter.	Image will be Uploaded.	Success
4	GUI	It will open a GUI application.	Window will be open.	Window Displayed.	Success
5	Test Image	We need to provide the Test Image Conditions.	The recognition of the image is done Successful.	The Image is Recognized.	Success
6	Face recognition	Image depicting a face.	Face is recognized	Face is properly recognized	Success
7	Compare the database	Face features are checked in database	Compared the face in database	Face compared successfully	Success
8	Output Screen	The output of Test Image will be displayed.	Users face should be recognized.	User face is Recognized.	Success

7. CONCLUSION AND FUTURE ENHANCEMENTS

7.1. CONCLUSION

An image processing and classification method has been implemented in which face images are used to train a dual classifier predictor. The predictor is relatively successful at predicting test data from the same dataset used to train the classifiers. This is likely due to a combination of lacking training and test images that clearly exhibit contempt, poor pre-training labeling of data, and the intrinsic difficulty at identifying contempt. The classifier is also not successful at predicting emotions for test data that have expressions that do not clearly belong exclusively to one of the seven basic expressions, as it has not been trained for other expressions. Future work should entail improving the robustness of the classifiers by adding more training images from different datasets, investigating more accurate detection methods that still maintain computational efficiency, and considering the classification of more nuanced and sophisticated expressions.

It can be used to solve intriguing tasks such as face recognition, although this task was quite convolute even more when using a great number of images. We humans also sometimes make a mistake while recognizing someone's so is our program. This system are important in day to day life, which take place in our daily lives. In Present Work, we recognize faces sing Python, OpenCV & Database and got some interesting insight about it.

7.2. FUTURE ENHANCEMENTS

Face recognition software differentiates a face from rest of the background in the image. The software first recognizes the face then measures different facial features. The software recognizes those features as nodal points.

- We recognizing the faces in the video Surveillance.
- Also providing the security in Banking and in Patient Identification in Health care.
- In Law Enforcement like Criminal Identification, Missing Persons.

8. REFERENCES

8.1. JOURNALS

[1] Yan Sun,Zhenyun Ren,and Wenxi Zheng "Face Recognition Algorithm Based on Image Processing", College of Information and Communication Engineering University, Harbin 150001, Heilongjiang, China,Mar 2022

[2] Joshila Grace.L.K and K. Reshmi, "Face recognition in surveillance system," 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2015, pp. 1-5, doi: 10.1109/ICIIECS.2015.7192887.

[3] G. Singh and A. K. Goel, "Face Detection and Recognition System using Digital Image Processing," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 2020, pp.348-352, doi: 10.1109/ICIMIA48430.2020.9074838.

8.2. BOOKS

- Introduction to Machine Learning with Python: A Guide for Data Scientists (1stEdition)
- Progress in Pattern Recognition, Image Analysis, Computer vision, and Applications:23rd Euromericans Congress, CIARP 2018, Madrid, Spain, November …. (Lecture Notes in computer Science, 11401) 1st ed.2019 Edition.
- Open cv 3.x with python By Example: Make the most of OpenCV and Python to build applications for object recognition and augmented reality, 2nd Edition 2nd Revised edition.

8.3. WEB LINKS

- https://ieeexplore.ieee.org
- https://stackoverflow.com
- http://www.studyglance.in

A Comprehensive Survey on the Face Recognition Using CNN

Mr. T. Madhu[*1] K.V. Hemanth[*2] Sheldon Shaji[*3] V. Nikil[*4]
Associate Professor[*1] *Scholar*[*2,3,4]
Department of Information Technology, Nalla Narsimha Reddy Education Society's Group of Institutions[*1,2,3,4]

Abstract

Face recognition is one of the interesting research topics in the field of computer vision. In recent years, deep learning methods, especially the Convolutional Neural Network, have progressed. One of the successes of CNN is in face recognition. Face recognition by computer is a technique done so that the computer can automatically recognize faces in an image. This survey presents research related to face recognition based on Convolutional Neural Network that has been conducted. The studies used are studies that have been published in the last five years. It was performed to determine the renewal that emerged in face recognition based on Convolutional Neural Network. The basic theory of the Convolutional Neural Network, face recognition, and description of the database used in various researches are also discussed. Hopefully, this survey can provide additional knowledge regarding face recognition based on the Convolutional Neural Network. Keywords: convolutional neural network; deep learning; face recognition; survey.

Introduction

Humans often use faces to recognize individuals and advancements in computing capability over the past few decades now enable similar recognitions automatically. Early face recognition algorithms used simple geometric models, but the recognition process has now matured into a science of sophisticated mathematical representations and matching processes. Major advancements and initiatives in the past ten to fifteen years have propelled face recognition technology into the spotlight. Face recognition can be used for both verification and identification (open-set and closed-set). In face recognition system it identifies faces present in the images and videos automatically. It is classified into two modes:

1. Face verification (or authentication)
2. Face identification (or recognition)

Face recognition has been an active research area over last 40 years. The face recognition research has several disciplines such as image processing, machine learning approach, pattern recognition, computer vision, and neural networks. Classification is the main problem. In the process of face recognition it includes, to train the face images from the known individuals and then to classify the newly coming test images into one of the classes

In automatic face recognition system, the main complicated task is that it involves detection of faces from a cluttered background, facial feature extraction, and face recognition. A complete face recognition system has to solve all sub-problems, where each one is a separate research problem. Image template based and geometry feature-based are the two classes of face recognition system algorithms. In template-based method it (Robert J. 1981) computes the correlation between a face image and one or more model of face image templates to estimate the face image identity from the database. Brunelli and Poggio (R. Brunelli, 1993) suggest the optimal strategy for face recognition system which is holistic and corresponds to template matching. The statistical tools such as Support Vector machines (SVM) (E. Osuna, 1997), (Vladimir N, 1995) Independent component Analysis, Principal Component Analysis (PCA) (L. Siro ich, 1987), (Matthew Turk, 1991), Linear Discriminant Analysis (LDA) (Peter N. Belhumeur et.al, 1997), used to construct a suitable database of face image templates.

LITERATURE SURVEY

Zafar et al. focused on making Bayesian DCNN for face recognition [1]. The researchers formed three CNN models with different depths and a Bayesian-DCNN. The use of different models is done to test the accuracy of each model. The dataset used in the study is the AT&T Face Database When tested with 100% with the AT&T Face Database.

Bendjillali et al. in their research, is performed with the architecture Inception-v3 [2]. The Viola-Jones algorithm is used to detect the face. The authors increased the contrast of the training image to determine the impact given to the accuracy of face recognition. Testing is done using CMU PIE datasets. The results obtained in testing using the CMU PIE, the accuracy of face recognition with Inception-v3 is 99.89%.

Hu et al. used 3 Convolutional Neural Networks arranged in parallel and proposed a Diversity Combination method for face recognition [3]. Inception-ResNet-v1 was used as the basic architecture of CNN. The three CNNs were used for feature extraction, and diversity combination is a strategy used to adaptively adjust the weight value in each CNN and make joint classification decisions. CAISA Web Face were used to train the CNN. Face matching testing was performed using CASIA NIR-VIS 2.0 and Oulu-CASIA NIR-VIS dataset. The accuracy level obtained was 99.8% using the Oulu-CASIA NIR-VIS dataset.

Li et al. in their research used multi-CNN [4]. Detection of facial components (eyes, eyebrows, nose, mouth) is done using the Active Shape Model (ASM). CNN is then used to extract features from each facial component. The dataset used for training and testing is Nenu LD. The highest level of accuracy is achieved is 99.5%.

Ding proposed a CNN model called Trunk-Branch Ensemble CNN (TBE -CNN) for video-based face recognition [5]. The model was designed to be able to extract additional information from the face holistically, and facial parts took around facial components. The TBE-CNN model is based on Go o g LeNet. Data augmentation. Test are performed by COX Face; accuracy was achieved using the Pa SC database 99.33%.

Deng et al. proposed a facial recognition algorithm to recognize a masked face based on large margin cosine loss called MF Cos face [6]. In their work, MTCNN was utilized for face detection, and the base architecture used is the Inception-ResNet-v1, and the proposed large margin cosine loss was used to train the model. Some faces from the datasets are generated with a mask. The accuracy achieved on LFW m is 99.33% respectively.

Yang et al. proposed a face matching method named SR-CNN [7]. The study combines Rotation- Invariant Texture Features (RITF) and Convolutional Neural Network (CNN). Testing and training are done using the LFW dataset. The results obtained reached 98.98%. The combination of RITF, SIFT, and CNN provides an increase in the level of accuracy of face matching.

Nam et al. proposed a Pyramid-Based Scale-Invariant Convolutional Neural Network (PSI-CNN) model [8]. The method is used to overcome the problem of input image scale, for CNN to recognize faces on low-resolution images, therefore increasing the performance of CNN. Evaluation and training is done using the LFW database, which is a standard dataset for face recognition evaluation. The results obtained are that the model has a face matching accuracy level of 98.87%.

Peng et al. introduced a CNN called NIR Face Net [9]. NIR Face Net is a modification of Goo g Le Net, used for face recognition whose input is in the form of Near-Infrared (NIR) images. The researchers chose to do face recognition with a NIR image because it has advantages over lighting changes. The dataset used is the CASIA NIR database. Then, the testing with blurred and noise images achieved accuracy ranging from 96.02% to 98.48%.

Nam et al. proposed a Pyramid-Based Scale-Invariant Convolutional Neural Network (PSI-CNN) model [10]. The method is used to overcome the problem of input image scale, for CNN to recognize faces on low-resolution images, therefore increasing the performance of CNN. Evaluation and training is done using the LFW database, which is a standard dataset for face recognition evaluation. The results obtained are that the model has a face matching accuracy level of 98.37%.

Pei et al. made a student attendance system using face recognition based on deep learning [11]. There was a difficulty in getting a large amount of training data. Data augmentation was used as a solution to increase the number of images in the dataset that can be used. Researcher used the CNN VGG-16 architecture had taken data through face capture via video, and the level of accuracy increased to 98.1.

Khan et al. used a Convolutional Neural Network to detect and recognize faces [12]. R-CNN that is used for object detection, is used by researchers to detect faces in images. The dataset used for training is LFW, and the authors used the data augmentation method by reversing each image in the dataset. The accuracy achieved at around 97.9%.

Song et al. proposed the Spartan Face Mask Detection and Facial Recognition system to address the challenge of mask detection and identity recognition that emerged during the Covid-19 pandemic [13]. The proposed system utilized the MTCNN to detect faces in an image, Face Net to extract embedded facial features, The training and testing were done using a total of 2000 images. The accuracy achieved using the Face Net + SVM is 97%

Ben Fredj et al. trained a CNN to recognize faces in an uncontrolled environment, in the sense that face images have noise, or partially covered faces [14]. The researchers used the data augmentation method of flipping. Data augmentation increases the number of images in the dataset and adds variation. The research in [19] has little in common with research. when tested with the YTF dataset it gives a result of 96.83%.

Hu et al. proposed a method for increasing the number of images in the dataset through image synthesis [15]. The architecture is named CNN-L. CNN-L has more layers than CNN-S The dataset used for training is NIR-VIS 2.0 The test results show the highest accuracy obtained using CNN-L, which is 95.77% The model was designed to be able to extract additional information from the face holistically, and facial parts took around facial components.

Ullah et al. proposed a unified framework for mask detection and masked facial recognition [16]. The authors made a custom CNN called Deep Mask Net, which comprises 17 layers, including the input and classification layers. was used for training and testing the model. The Deep Mask Net achieved 100% accuracy in face mask detection and 93.33%.

Chen et al. in their study, proposed a CNN with deep transformation learning [17]. The method increases the robustness and degree of discrimination of the extracted features. The datasets used for training are CASIA-Web face. Tests were carried out by researchers using the LFW and IJB- A datasets. The test results show that the accuracy rate obtained using IJB-A is 93.1%.

Zangeneh et al. [18] uses the model based on the VGG net model. Model training uses the UMIST and FERET datasets. Testing is done using FERET face datasets. The accuracy obtained using the FERET dataset reached 92.1%.

Moon et al. trained CNN to recognize a person's face at different distances [19]. The used dataset is a dataset formed by the researchers themselves (private), with 12 individuals, and 300 images for each individual. Everyone's faces in the dataset are taken from 5-meter to 15 meters, with 30 images taken at each distance. The average level of accuracy obtained from this study was 88.9%.

Deep Learning Methods for Feature Extraction

Deep Learning:

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. The Figure (a) discusses about Deep Learning. These neural networks attempt to simulate the behaviour of the human brain albeit far from matching its ability allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy. Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services as well as emerging technologies.

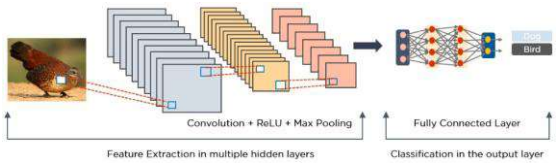


Fig (a): Deep Learning

OPEN FACE:

Open Face is an open-source tool intended for computer vision and machine learning researchers, the affective computing community and people interested in building interactive applications based on facial behaviour analysis. It is the first open-source tool capable of facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation. The computer vision algorithms which represent the core of Open Face demonstrate state-of-the-art results in all of the above-mentioned tasks.

ARC FACE:

Arc Face is a specific approach or loss function used in face recognition tasks. The figure (b) describes the how the arc face will work. It's designed to enhance the discriminative power of features extracted from facial images, improving the accuracy of face recognition systems. Arc Face is particularly known for its effectiveness in handling the challenging problem of large variations in lighting, pose, and facial expressions.

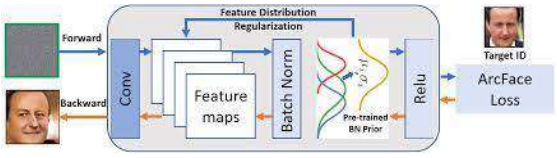


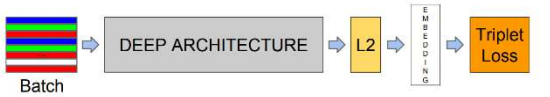
Fig (b): Arc Face

VGG FACE:

There are many pre-trained deep learning models so that by using those architectures we can train our model instead of creating a new architecture every time It is based on the VGG architecture, which is known for its deep convolutional neural network (CNN) layers. The VGG Face model is capable of extracting rich and discriminative features from facial images, enabling accurate face recognition. Multiple layer detection on the last feature map is used to detect varied face sizes. Details of how to crop the face given detection can be found in vgg face misconnect package below in class face Crop.

FACE NET:

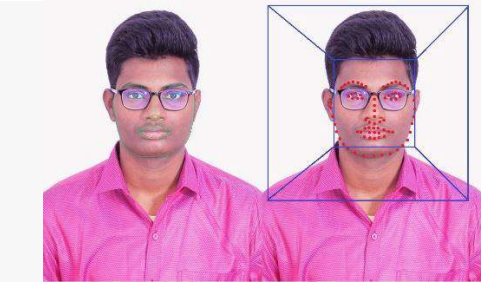
Face Net is a deep learning model for face recognition developed by researchers at Google. The below fig (C) explains the working of Face net. It's designed to directly learn a mapping from facial images to a compact Euclidean space, where the distance between points in this space corresponds to the similarity between the faces. Face Net leverages a triplet loss function to learn discriminative face embeddings that can be used for accurate face recognition and verification tasks.



Fig(c): Face Net

Dlib Face Recognition:

Dlib is an open-source suite of applications and libraries written in C++ under a permissive Boost license and algorithms for various computer vision tasks, including face recognition. It's known for its efficiency and effectiveness through Dlib also features robust tools for object pose estimation, object tracking, face detection and face recognition (identifying a perceived face). Dlib's face recognition capabilities are built upon deep learning techniques and classic machine learning algorithms. The below fig (d) describes about the Dlib's Face Recognition.



Fig(d): Dlib Face Recognition

DEEP ID:

"Deep ID" refers to a series of deep learning-based face recognition models developed by the Chinese University of Hong Kong (CUHK). Deep ID model has shown very promising results for 2D face recognition, Multi-task learning face recognition and attribute prediction does not improve performance. Deep ID models aim to learn compact and discriminative representations of faces, it seems reasonable to somehow integrate parts of this model in our general model. The below fig (e) describes about the Deep ID.

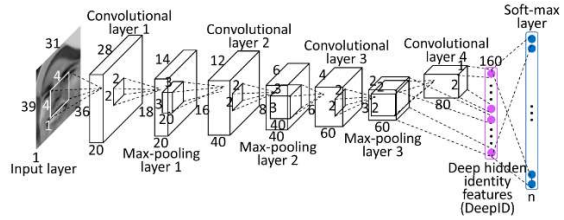


Fig (e): Deep ID

Table 1: Advantages and Disadvantages of Deep Learning Methods

Methods	Advantages	Disadvantages
Deep Learning	High Accuracy, Adaptability and Real-Time Processing	Data Requirements, Computational Resources
Open Face	Trained models, Scalability	Maintenance, Complexity and the Resource Intensive
Arc Face	Variations, Performance	Data Requirements, Complexity and Computationally Intensive.
VGG Face	Features, Transfer Learning	Efficiency, Limited Context and Computational Requirements
Face Net	Variations, Learning	Parameter Tuning, Training Data and Computational Demands
Dlib Face Recognition	Efficiency, Accuracy, Pre-Trained Models	It does not detect small faces in the image
Deep Id	Variations, Recognition, Pre-Trained Models	Over Fitting

The Table-1 tells us about the advantages and Disadvantages of the Deep Learning Methods.

Table 2: Based on the CNN- Based Face Recognition Research

Research	CNN Model	Face Database	Accuracy
Zafar et al[1]	B-DCCN	AT&T Face Database	100%
Bendjillali et al[2]	Inception-v3	CMU PIE database	99.89%.
Hu et al[3]	Inception-ResNet-v1	Oulu-CASIA NIR-VIS dataset	99.8%
Li et al[4.]	Multi-CNN	Nenu LD.	99.5%
Ding [5]	(TBE -CNN)	Pa SC database,	99.33%
Deng et al[6]	Inception-ResNet-v1	L F Wm	99.33%
Yang et al[7]	SR-CNN	LFW database	98.98%.
Nam et al [8]	(PSI-CNN)	LFW database	98.87%.
Peng et al[9]	Google Net	CASIA NIR database	98.48%
Nam et al[10]	(PSI-CNN)	LFW database and YTF dataset	98.37%.
Pei et al[11]	CNN VGG-16	Private	98.1%
Khan et al[12]	R-CNN	LFW database	97.9%.
Song et al [13]	Face-Net + SVM	Private	97%
Ben Fredj et al[14]	CNN	YTF dataset	96.83%
Hu et al[15]	CNN-L and CNN-S	NIR-VIS 2.0	95.77%
Ullah et al[16]	Deep Mask Net	MDMFR	93.33%
Chen et al[17]	CASIA-Web face	LFW and IJB- A database	93.1%
Zangeneh et al[18]	VGG-net model	UMIST and FERET datasets.	92.1%.
Moon et al[19]	CNN	Private	88.9%

The Table-2 shows the different researchers about the CNN Based Face recognition and also their accuracy by using different CNN model and Datasets.

Conclusion

This survey discusses face recognition based on the Convolutional Neural Network. Face recognition is one of the challenges in pattern recognition and computer vision and one of the many studies conducted in recent years. Several studies have been discussed trying to find renewal for face recognition. The renewed face recognition proposed various kinds of architecture used, modifying the images in the dataset or combining several methods. The main goal is to obtain a high level of accuracy so that the face recognition system has a high performance. This survey also discusses some face databases used in several studies. The face databases available have data from hundreds of images to millions. It is hoped that through this survey, readers can gain additional knowledge about face recognition based on Convolutional Neural Networks.

References

[1] U. Zafar et al., “Face recognition with Bayesian convolutional networks for robust surveillance systems,” EURASIP J. Image Video Process., vol. 2019, no. 1, p. 10, Dec. 2019.

[2] R. I. Bendjillali, M. Beladgham, K. Merit, and A. Taleb-Ahmed, “Illumination-robust face recognition based on deep convolutional neural networks architectures,” Indones. J. Electr. Eng. Comput. Sci., vol. 18, no. 2, p. 1015, May 2020.Loss and Diversity Combination,” IEEE Access, vol. 7, pp. 75305–75317, 2019.

[3] G. Hu, X. Peng, Y. Yang, T. M. Hospedales, and J. Verbeek, "Frankenstein: Learning Deep Face Representations Using Small Data," IEEE Trans. Image Process., vol. 27, no. 1, pp. 293–303, 2018.

[4] P. Li, J. Xie, W. Yan, Z. Li, and G. Kuang, "Living Face Verification via Multi-CNNs," Int. J. Comput Intell. Syst., vol. 12, no. 1, p. 183, 2018

[5] C. Ding and D. Tao, "Trunk-Branch Ensemble Convolutional Neural Networks for Video-Based Face Recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 1002–1014, 2018.

[6] H. Deng, Z. Feng, G. Qian, X. Lv, H. Li, and G. Li, "MFCosface: a masked-face recognition algorithm based on large margin cosine loss," Appl. Sci., vol. 11, no. 16, 2021.

[7] Y. X. Yang, C. Wen, K. Xie, F. Q. Wen, G. Q. Sheng, and X. G. Tang, "Face recognition using the SR-CNN model," Sensors (Switzerland), vol. 18, no. 12, 2018.

[8] G. P. Nam, H. Choi, J. Cho, and I. J. Kim, "PSI-CNN: A Pyramid-based scale-invariant cnn architecture for face recognition robust to various image resolutions," Appl. Sci., vol. 8, no. 9, 2018.

[9] M. Peng, C. Wang, T. Chen, and G. Liu, "NIRFaceNet: A Convolutional Neural Network for Near-Infrared Face Identification," Information, vol. 7, no. 4, p. 61, Oct. 2016.

[10] G. P. Nam, H. Choi, J. Cho, and I. J. Kim, "PSI-CNN: A Pyramid-based scale-invariant cnn architecture for face recognition robust to various image resolutions," Appl. Sci., vol. 8, no. 9, 2018.

[11] M. Z. Khan, S. Harous, S. U. Hassan, M. U. Ghani Khan, R. Iqbal, and S. Mumtaz, "Deep Unified Model for Face Recognition Based on Convolution Neural Network and Edge Computing," IEEE Access, vol. 7, pp. 72622–72633, 2019.

[12] G. P. Nam, H. Choi, J. Cho, and I. J. Kim, "PSI-CNN: A Pyramid-based scale-invariant cnn architecture for face recognition robust to various image resolutions," Appl. Sci., vol. 8, no. 9, 2018.

[13] Z. Song, K. Nguyen, T. Nguyen, C. Cho, and J. Gao, "Spartan Face Mask Detection and Facial Recognition System," Healthc., vol. 10, no. 1, 2022.

[14] H. Ben Fredj, S. Bouguezzi, and C. Souani, "Face recognition in unconstrained environment with CNN," Vis. Comput., no. 0123456789, Jan. 2020.

[15] Z. Pei, H. Xu, Y. Zhang, M. Guo, and Y.-H. Yang, "Face Recognition via Deep Learning Using Data Augmentation Based on Orthogonal Experiments," Electronics, vol. 8, no. 10, p. 1088, Sep. 2019

[16] N. Ullah, A. Javed, M. Ali Ghazanfar, A. Alsufyani, and S. Bourouis, "A novel DeepMaskNet model for face mask detection and masked facial recognition," J. King Saud Univ. - Comput. Inf. Sci., no. 2022.

[17] G. Chen, Y. Shao, C. Tang, Z. Jin, and J. Zhang, "Deep transformation learning for face recognition in the unconstrained scene," Mach. Vis. Appl., vol. 29, no. 3, pp. 513–523, Apr. 2018.

[18] E. Zangeneh, M. Rahmati, and Y. Mohsenzadeh, "Low resolution face recognition using a two-branch deep convolutional neural network architecture," Expert Syst. Appl., vol. 139, p. 112854, 2020.

[19] H.-M. Moon, C. H. Seo, and S. B. Pan, "A face recognition system based on convolution neural network using multiple distance face," Soft Comput., vol. 21, no. 17, pp. 4995–5002, Sep. 2017.