FACE DECTECTION AND RECOGNIZATION

**PROJECT SYNOPSIS**

OF MINOR PROJECT

**BACHELOR OF TECHNOLOGY**

CS DS & 4TH

SUBMITTED BY

**ARIF ALAM ANSARI, AMRITANSHU SHARMA, ADITYA RAJ & ADIBA**

(2001331540036), (20013331540025), (20013331540011) & (20013331540006)



# STUDENT’S DECLARATION

I hereby certify that the work which is being presented in the major project report entitled “FACE DETECTION AND RECOGNIZATION” in fulfillment of the requirement for the award of the Degree of Bachelor of Technology in Department of Noida institute of engineering and technology greater Noida Uttar Pradesh is an authentic record of our own work carried out during 4th semester.

Date: Name and Signature of student

Arif

Amritanshu

Aditya

Adiba

The major project viva-voce examination of Mr./Ms Arif, Amritanshu, Aditya, Adiba Roll No. 2001331540036, 25, 11, 06 Of B. TECH ( Branch) has been held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of:

Project Guide:

Head of Dept: (Stamp of organization)

**ABSTRACT**

While recognizing any individual, the most important attribute is face. It serves as an individual identity of everyone and therefore face recognition helps in authenticating any person`s identity using his personal characteristics. The whole procedure for authenticating any face data is subdivided 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. Then the whole process is repeated thereby helping in developing a face recognition model which is considered to be one of the most extremely deliberated biometric technology. 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 uses PCA (Principal Component Analysis) to minimize the facial dimension space of facial features. The concern of this work is the use of digital image processing for the development of facial recognition systems.

**ACKNOWLEDGEMENT**

We are highly grateful to the Dr. , Principal, **Noida institute of engineering and technology, Greater Noida Uttar Pradesh** for providing this opportunity.

The constant guidance and encouragement received from , HOD ( , dept.), NIET, Greater Noida has been of great help in carrying out the project work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to \_ project guide, without the wise counsel and able guidance, it would have been impossible to complete the report in this manner.

We express gratitude to other faculty members of CSE-DS department of GIMT for their intellectual support throughout the course of this work.

Finally, the authors are indebted to all whosoever have contributed in this report work.

Arif Alam Ansari

Adiba

Amritanshu Sharma

Aditya Raj

### 

FIGURE INDEX

Fig No. Page No.

1 ……………………………………………………………………………… ix

2 ……………………………………………………………………………… ix

3 ……………………………………………………………………………… x

4 ……………………………………………………………………………… xi

5 ……………………………………………………………………………… xii

6 ……………………………………………………………………………… xxii

7 ……………………………………………………………………………… xxiii

8 ……………………………………………………………………………… xxiv

9 ……………………………………………………………………………… xxv

10 ……………………………………………………………………………… xxv

11 ……………………………………………………………………………… xxviii

12 ……………………………………………………………………………… xxix

13 ……………………………………………………………………………… xxx

14 ……………………………………………………………………………… xxxi

15 ……………………………………………………………………………… xxxii

16 ……………………………………………………………………………… xxxii

17 ………………………………………………………………………………

CONTENTS

**Chapter 1- Introduction**

1.1 Objective

1.2 Problem definition

1.3 Scope

1.4 Definitions, Acronyms and Abbreviations

1.5 Technologies to be used

**Chapter 2- Software Requirement Specifications**

2.1 Introduction

2.1.1 Purpose

2.1.2 Project Scope

2.2 Overall Description

2.2.1 Product/Project Perspective

2.2.2 Product/Project Function

2.2.3 User Classes and Characteristics

2.2.4 Operating Environment

2.2.5 Architecture Design

2.2.6 Constraints

2.2.7 Use Case Model Description

2.2.8 Assumptions and Dependencies

2.3 System Features

2.4 External Interface Requirements

2.4.1 User Interfaces

2.4.2 Hardware Interfaces

2.4.3 Software Interfaces

2.4.4 Communications Interfaces

2.5 Other Nonfunctional Requirements

2.5.1 Performance Requirements

2.5.2 Safety Requirements

2.5.3 Security Requirements

2.5.4 Software Quality Attributes

**Chapter 3- System Design**

3.1 Flowcharts

3.2 ER Diagram

3.3 Design Methodology

3.4 Software Development Model

3.5 Database Design

3.5.1 ER Diagram

3.5.2 Schema

3.6 DFD’s

3.7 Activity Diagram

3.8 Class Diagram

3.8.1 Sequence Diagrams

**Chapter 4- System Implementation**

4.1 Coding

4.2 Testing

4.3 Snapshots

**Chapter 5- Conclusions and future scope**

5.1 Conclusion

5.2 Future Scope

**Chapter 6- References**

6.1 Books

6.2 URLs

**Chapter 1- Introduction**

1.1 Objective

The objective of human activity recognition (HAR) is to identify activities based on a series of observations of people' behaviours and ambient variables. Many applications, like as video surveillance, health care, and human-computer interaction (HCI), are founded on vision-based HAR research. The advancements of state-of-the-art activity recognition systems are highlighted in this study, particularly for activity representation and classification methods. We sort out a chronological research trajectory from global representations to local representations, as well as contemporary depth-based representations, for the representation approaches. We categorise template-based approaches, discriminative models, and generative models as classification methods, and we discuss various popular methods.

1.2 Problem definition

The task of recognizing human activity in general is quite broad. As a result, the field has been conceptually divided, and we typically talk about action primitives, actions, activities, interactions, and group activities, depending on the complexity of the activity. Frequently, the activities and interactions involve a complex visual scenario with additional objects. Each of these categories necessitates a unique approach to the issue. Identify or distinguish one or more people in the scene using a stored database of faces. In a traffic video, recognize a person's face. The gadgets are controlled by facial expression and eye movement. As an example, Attendance Management System is software that is used in schools, colleges, and institutes to track daily student attendance. It makes it easier to find out about a specific student's attendance in a specific class. This system will also assist in determining a student's attendance eligibility criteria. The system will be able to generate the students' attendance report with a single mouse click, reducing the need for manual labor, which is prone to human error and time consuming.

1.3 Scope

As the world moves toward an internet platform, some advancements in technology can benefit in the fields of education, surveillance, and medicine.



Fig 1

Passports and visas may also be verified using this technology. A driver's license can also be verified using the same manner. In the defense ministry, airports, and other vital areas, technology might be used to increase surveillance and security. It might also be used to identify candidates during exams. In a number of government and business contexts, this approach may be used to verify and track attendance. It might also be used in bank lockers and vaults to verify and authenticate genuine users for access control. Police agencies may be able to use the approach to identify criminals.

1.4 Definitions, Acronyms and Abbreviations

*Definition 1* OpenCVis a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more.

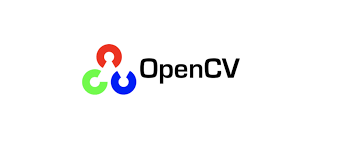


Fig 2

*Definition 2* Human activity recognition (HAR) The difficulty of identifying sequences of accelerometer data captured by specialized harnesses or smart phones into recognized well-defined motions is known as human activity identification. Activity can be broken down into four categories: gestures, interactions, acts, and group activities.



Fig 3

Gesture: A gesture is a visible body activity that conveys a message. It Okay gestures

and thumbs up are examples of movements made with the hands, face, or other areas

of the body. Rather of verbal or vocal communication.

• Action: The term "action" refers to a set of physical actions carried out by a single

person, such as walking or running.

• Interactions: Interactions are a collection of actions carried out by No more than two

actors. One of the subjects the first must be a person, whereas the second might be

either a human or an item (hand shaking, chatting, etc.).

• Groups activities: It's a blend in a group environment of gestures, movements, and

interactions There must be at least two actors, as well as one or more interactive

objects (playing volleyball, obstacle racing, etc.).

***Acronyms and Abbreviations***

Human activity recognition (HAR)

Human-computer interaction (HCI)

Open-Source Computer Vision Library as OpenCV

Entity Relationship Diagram as ER Diagram

1.5 Technologies to be used

It is possible to make use of OpenCV is a fantastic image processing and computer vision tool. It's a free and open-source initiative. a library that can be used to perform tasks like face recognition, objection tracking, landmark detection, and much more. It works with a variety of programming languages, including Python, Java, and C++, to name a few.

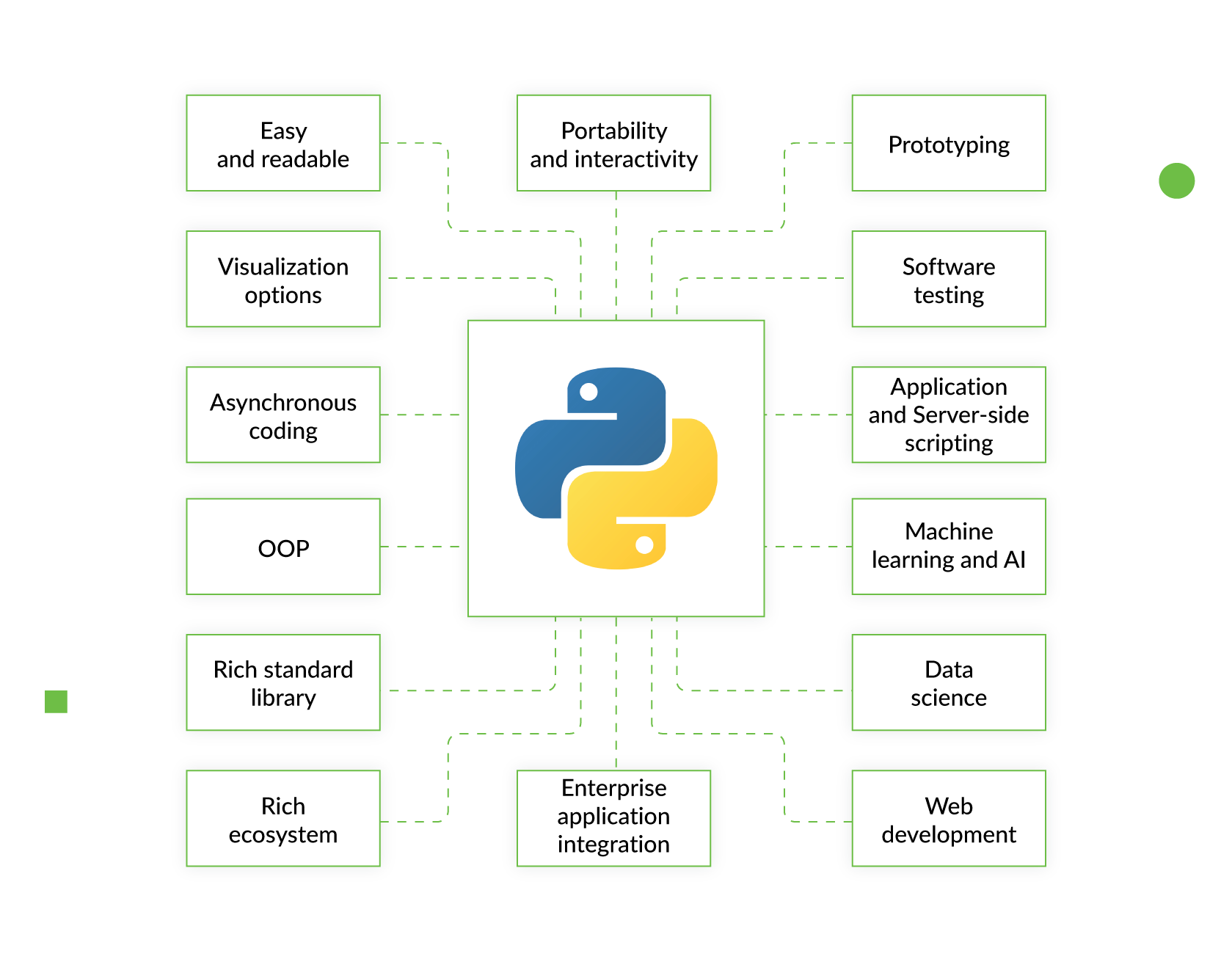


Fig 4

The Raspberry Pi is a small, inexpensive computer the size of a credit card. a computer with a monitor or television attached to it standard keyboard and mouse are included with the computer. Scikit-image is a free, open-source image processing library. Python is a programming language that can be used to create programmes. Geometric algorithms, segmentation algorithms, colour space manipulation, analysis, and transformations are all part of it. Filtering, morphology, and so on.

By using OpenCV to do our work (Open-Source Computer Vision Library) The main benefit of OpenCV is its large algorithm library. algorithmic efficiency, as well as a wide range of applications OpenCV's main characteristics are as follows: Programming Functions Library for Multiple Interfaces Public Discussion Forums.

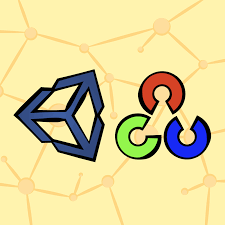


Fig 5

AI plays a critical part in the success of this endeavor. For example, recognizing pictures from live photographs and videos. AI will assist in determining if a pupil is napping or concentrating. In order to find the pattern in each image. For the purpose of comparing the data to information already kept in our database.

**Chapter 2 - Software Requirement Specifications**

2.1 Purpose:

Face Recognition is a technology in computer vision. In Face recognition / detection we locate and visualize the human faces in any digital image.

It is a subdomain of Object Detection, where we try to observe the instance of semantic objects. These objects are of particular class such as animals, cars, humans, etc. Face Detection technology has importance in many fields like marketing and security

use these pre-trained Haar cascade models to detect Human Face. We will implement a real-time human face recognition with python

Now we have seen our algorithms can detect faces but can we also recognize whose faces are there? And what if an algorithm is able to recognize faces?

Generally, Face Recognition is a method of identifying or verifying the identity of an individual by using their face. Various algorithms are there for face recognition but their accuracy might vary. Here I am going to discuss with you that how we can do face recognition using deep learning.

Now let us understand how we can recognize faces using deep learning. Here we use face embeddings in which every face is converted into a vector. The technique of converting the face into a vector is called deep metric learning.

What is the scope of facial recognition?

Though there are some weaknesses of facial recognition system, there is a tremendous scope in India. This system can be effectively used in ATM’s, identifying duplicate voters, passport and visa verification, driving license verification, in defense, competitive and other exams, in governments and private sectors.

2.2 Project perspective/product

Intel's OpenCV is a free and open-access image and video processing library. It is linked to

computer vision, like feature and object recognition and machine learning. This paper presents the main OpenCV modules, features, and OpenCV based on Python. The paper also presents common OpenCV applications and classifiers used in these applications like image processing, face detection, face recognition, and object detection. Finally, we discuss some literary reviews of OpenCV applications in the fields of computer vision such as face detection and recognition, or recognition of facial expressions such as sadness, anger, happiness, or recognition of the gender and age of a person.

OpenCV Library. It is a massive open-source image processing, machine learning, and computer vision library. OpenCV is compatible with a broad range of programming languages, including Python, C++, and Java. It will analyze photographs and videos to recognize artifacts, faces, and even human handwriting. When paired with many other libraries, like NumPy, a high-performance library for turning machines, achieve a good performance; that is, all services that can be performed in NumPy can also be integrated with OpenCV. It is written based on C++ and has a C++ interface as its main interface, but it also has a less robust but still detailed older Language training. Both the latest technologies and algorithms are visible in the C++ GUI. Python, Java, and MATLAB/OCTAVE bindings are available. Wrappers in a variety of programming languages have been created to promote broader acceptance. JavaScript plugins for a variant of OpenCV functions are published as OpenCV.js in version 3.4, which can be used on web platforms. The OpenCV project, which was officially unveiled in 1999, was originally Intel’s research program to support CPU-intensive applications. OpenCV is a popular platform for implementing face detection and recognition algorithms. The following are some often used OpenCV algorithms.

Haar Cascade Haar Cascade is an effective method for detecting objects. It’s a machine-learning-based method in which a cascade of actions is learned from a large number of positive and negative images. It becomes used to seeing things in different frames. shows the view of the Haar cascade classifier. shows the Haar cascade flowchart.

LBP (Local Binary Pattern) It is a simple but effective texture operator that labels pixels in an image by thresholding the pixels' neighborhood and treating the result as a binary number. The LBP texture operator has become a common approach in a variety of applications due to its discriminative power and computational simplicity. It can be viewed as a unifying solution to texture analysis's historically divergent statistical and structural models. The LBP operator's robustness to monotonic grayscale changes induced, for example, by illumination variations is perhaps its most significant property in real-world applications. Another key feature is its computational simplicity, which allows it to analyze images in difficult real-time scenarios. shows the description of facial expressions with local binary patterns Eigenfaces.

It is a method that employs PCA (Principal Component Analysis) to minimize dimensionality and find the strongest vectors for distributing facial images onto existing facial spaces. The primary goal of PCA is to identify the best vectors to explain the distribution of facial images in picture space into face space. According to the eigenvalue distribution, m eigenvector is used to construct the principal component amount. Eigenvector and eigenvalue are calculated from the qualified facial image's covariance matrix. The eigenvector is sorted by eigenvalue (high to low) and M first eigenvectors are chosen to form the principal variable Fisher Faces. It is a face recognition system that several researchers have demonstrated to identify faces accurately. Fisher Face is a calculation model that combines the PCA (Principal Component Analysis) calculation model with Fisher’s Linear Discriminant (FLD). PCA is used to minimize input data in order to simplify and accelerate the FLD operation. On the other hand, FLD is used to generate a distribution matrix to aid in classification and identification. A series of Fisher Faces is generated using the PCA and FLD calculation models. This facial recognition process consists of four major steps: face identification, PCA estimation, calculation, and classification. LBPH (Local Binary Pattern Histogram) LBP is a highly effective texture operator. That compares each neighboring pixel's threshold value to the value of the center pixel. It takes into account outcomes in the context of binary numbers. LBP is a common technique in a variety of applications due to its discriminative strength and simplicity. LBP was identified for the first time in 1994. Since then, it seems to have evolved into a more efficient texture classification algorithm. It was later discovered that combining LBP with histograms of directed gradient descriptors increases its accuracy on the same dataset. LBP has additional capabilities such as monotonic grey-scale improvements and statistical simplicity, allowing it to interpret images in real-time applications. 4 shows the LBPH algorithm for face recognition

YOLO is an abbreviation for (you only look once). It is the most recent real-time object detection system that uses a single neural network to process the entire image. This network splits the picture into sections and estimates the bounding boxes and probabilities for each. The estimated probabilities are used to weigh these bounding boxes. The testing phase examines the whole picture, so the image's global meaning guides its predictions. Can detect objects form videos or image shows the Yolo process.

Faster R-CNN It is a popular object detection architecture proposed in 2015 by Ross\_Girshick and is one of the most well-known object detection architectures that employ convolution neural networks. The implementation of the Region Proposal Network enables (Faster R-CNN) easier and quicker (RPN). RPN is a completely convolutional network that has been trained side-to-side, and forecasts object boundaries and object ratings at each detection. Since RPN is so critical to (Faster-R-CNN) and remains one of the strongest entity detection frameworks open to researchers, most of this piece would concentrate on RPN architecture and the notions of anchor boxes and suppression non-maximum. shows the Faster\_ R-CNN step

Single Shot Detectors (SSDs) The SSD method is focused on the feed-forward convolutional network that generates a permanent border-box array and results in the existence of class-based entity instances in these boxes and a non-maximum deletion stage to generate final detection. The early network layers are built on a common image classification design shows the SSD.

Fast face processing. Our biometric face recognition system performs fast and accurate detection of face in live video stream. The face on the current frame is detected in 0.07 sec. and then each face is processed in 0.13 sec.

Live face detection. A conventional face identification system can be easily cheated by placing a photo of another person in front of a camera. Our face recognition system is able to prevent this kind of security breach by determining whether a face in a video stream belongs to a real human or is a photo

Face image quality determination. A quality threshold can be used during face enrolment to ensure that only the best quality face template will be stored into database.

Tolerance to face posture. Our face recognition system has certain tolerance to face posture that assures face enrolment convenience: rotation of a head can be up to 10 degrees from frontal in each direction (nodded up/down, rotated left/right, tilted left/right).

Multiple samples of the same face. Biometric template record can contain multiple face samples belonging to the same person. These samples can be enrolled with different face postures and expressions, from different sources and in different time thus allowing to improve matching quality.

Identification capability. Our system functions can be used in 1-to-1 matching (verification), as well as 1-to-many mode (identification).

Fast face matching. The JUSTLOOK 3.2 face template matching algorithm compares 100,000 faces per second.

Compact face features template. A face features template occupies only 2.3 Kilobytes; thus our applications can handle large face databases.

Features generalization mode. This mode generates the collection of the generalized face features from several images of the same subject. Then, each face image is processed, features are extracted, and the collections of features are analyzed and combined into a single generalized features collection, which is written to the database. This way, the enrolled feature template is more reliable and the face recognition quality

Extra Modules Extra modules are not included in the OpenCV release by default. Additional machine vision functionalities, such as text detection, are applied to these modules. The main modules are described below: ï‚· Core: Have most of OpenCV's core features. ï‚· Import: Picture processing tools such as transforms, manipulations, and filtering are used. ï‚· Miscodes: Image reading/writing features are included. ï‚· Video: Video reading/writing features are included. ï‚· High-up: Used for creating GUI to display output. ï‚· Video: Motion detection and monitoring. ï‚· calib3d: This package includes calibration and 3D reconstruction functions for estimating translation between multiple pictures. ï‚· features2d: Item identification and categorization algorithms utilizing key point detection and descriptor extraction algorithms, which are included in this library. ï‚· Obj detect used for detecting object. ï‚· Dunn: Classify and detecting objects, between other objects. ï‚· Ml: Used for regression and classification, and it encompasses the vast majority of machine learning. ï‚· Flann: Supports optimized algorithms for searching for high-dimensional attributes in massive data sets using nearest neighbor search. Fast Library for Approximate Nearest Neighbors (FLANN) is an acronym for Quick Library for Estimated Nearest Neighbors (FLANN). Photo: Removes noise and creates HD images, among other roles for photography-related computer vision. ï‚· Stitching: Used for stitching image. ï‚· Shape: Deal with issues such as shape transformation, pairing, and distance

Library for Estimated Nearest Neighbors (FLANN). Photo: Removes noise and creates HD images, among other roles for photography-related computer vision. ï‚· Stitching: Used for stitching image. ï‚· Shape: Deal with issues such as shape transformation, pairing, and distance. ï‚· Supers: Deal with enhancement and resolution algorithms. ï‚· Video stab: Have algorithms for stabilizing video. ï‚· Viz: create 3D display window for widgets. hardware interface:

The Facial Recognition System (FRS) is implemented on Artix-7 XCA100T FPGA with 135.26 MHz clock frequency. It can recognize 5500 images per second with 98.75% accuracy on image of size 112 Ã— 92.

HARDWARE ACCELERATOR DESIGN

The face recognition algorithms implemented require matrix product operations which constitute the

"bottleneck" of the system, limiting the operation speed. So, it was decided to implement this part in hardware. The design of the hardware accelerator has been carried out using the high-level synthesis tool from Xilinx Viv ado-HLS [Xil14]. Viv ado-HLS Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and those copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. generates an RTL description starting from a C/C++ algorithm level description. It realizes the scheduling

and the resource allocation in order to map the algorithm to hardware. It also generates a description as IP module so that it can be used as a peripheral of a processor. This methodology allows the designer to start the design of the system to be implemented in hardware from high-level descriptions. This means that algorithmic descriptions are made in a high-level

programming language (in our case C++). VivadoHLS takes as input the high-level description and is able of generating a circuit that implements the desired algorithm. The designer can set restrictions, using directives (pragmas), on latency, throughput or hardware resources. As we will see, this methodology allows for exploration of the design space (due to automation), and design optimization through the application of directives.

software interface:

To achieve the face recognition in the Android platform, JNI (Java Native Interface) is used to call the local Open CV. The proposed system is tested in real-time in two different brands of smart phones, and results average success rate in both devices for face detection and recognition is 95% and 80% respectively.

Top 11 Facial Recognition Software in 2021:

Amazon Recognition Core services: Amazon Recognition is one of the most reliable names in the Facial recognition software game.

Facial analysis and facial search are used for user verification, people counting, and public safety use cases.

Recognition can identify objects and scenes by giving them labels.

Amazon has the advantage of having a massive dataset at its disposal. This ensures accuracy.

Recognition allows users to add custom labels to objects and scenes to suit business needs. This ensures higher match success rates. All the business has to do is supply images of the objects and locations to be identified.

Amazon provides content moderation. It has a tested dataset which it uses to flag inappropriate content. It also provides text detection to recognize names.

Its latest offering is PPE detection that identifies workers wearing PPEs for security.

It provides face search and verification within a private repository of photos too. Please note that storage costs are incurred in this case.

Recognition also provides face detection and analysis in videos â€” live or stored.

Beta face Core services: Beta face mainly focuses on image and video analysis and face and objection recognition.

It offers three kinds of services â€” facial recognition SDKs, customer software development services, and hosted web services.

Services can be simple face detection or complex facial recognition, identification, and verification.

It uses biometric measurements for facial feature tracking on both images and videos. It can recognize emotion and ethnicity.

It can also track skin, hair, facial features, and hairstyle shape.

It provides video surveillance and security software solutions

BioID Core services: BioID is a GDPR-compliant solution that provides biometrics-as-a-service. It provides cloud-based FRS services that can be accessed by your product using APIs. The software offers three products:

BioID web service: This is a SaaS offering that can be deployed on-premise or on the cloud.

Liveness detection: This is a recognition service to detect user presence using face, eye, and voice recognition. It is used to prevent online fraud and identity threats.

Photo Verify: This solution combines face detection technology with BioIDâ€™s Liveness Detection service to verify photos used as identity proofs

Cognatic Core services: Cognatic provides scalable and customizable FRS to customers through its open system architecture through â€˜FaceVacs.â€™ Cognatic offers five solutions:

FaceVACS-VideoScan ES Live: This can be used for recognizing faces in live video streams. BioID extends this solution by providing a way to count individuals, generate demographic statistics, and track people€™s flow.

FaceVACS-VideoScan ES: This is Cognatic€™s subscription-based enterprise solution to install and manage video scans. Cognatic selects the computing and camera equipment and manages it on-premises or on the cloud by deploying a Cognatic partner.

FaceVACS-DBScan ID: This is a biometric verification and identification solution.

FaceVACS-DBScan LE: This is a biometric verification solution for law enforcement.

FaceVACS-Entry: This solution integrates FRS software with state of the class hardware to create electronic gates in border control checkpoints.

Deep Vision AI Core services: Deep Vision AI provides FRS solutions for marketing and planning and for businesses looking to use facial verification for security.

It gathers the footfall data in a particular area of the city by recognizing age, gender, and ethnicity. This data is used to help advertisers and brands target customers with personalized ads. Facial recognition and verification for security.

Deep Vision€™s facial verification solution identifies individuals by users like law enforcement for security.

It provides a real-time analytics dashboard that can be customized.

Face++ Core services: Face++ provides four types of technology solutions:

Facial recognition for face detection, face comparison, and face search.

Human body recognition for body detection, skeleton detection, and body outline.

Image beautifies for merging faces in multiple photographs.

Image detection â€“for tagging faces on photographs.

Face First Core services: Face First aims to use DigitalID to replace cards and passwords. It mainly provides FRS-based solutions in four key areas:

Security solutions: These include authentication, access control, ID verification, and age verification.

Customer engagement: For loyalty programs and personalized ads.

Safety: For loss prevention and fraud mitigation with real-time alerting for identity spoofing attempts.

Business insights: For sentiment scoring, traffic analytics, audio analytics, and fraud analytics by businesses.

Kairos Core services: Kairos provides FRS-based web services and an SDK for businesses to integrate its solutions. It provides facial detection, identification, and verification services. The software can also track important demographic data. Its auto-tagging feature enables quicker search and indexing of images and videos.

Sense Time Core services: Sense Time provides face and body analyzing technology, besides its stand-alone FRS services. Its solutions boast high accuracy. It provides services like:

Face detection

Facial feature point positioning: Feature positioning is marked irrespective of wide-angles, changing expressions, or movement.

Facial attributes: The solution can accurately recognize more than ten facial attributes.

Liveness detection: User verification solution to prevent spoofing attacks.

Sky Biometry Core services: Sky Biometry is a web service provider which offers three primary services:

Face detection

Attribute determination

Facial recognition

Trueface.ai Core services: Trueface.ai provides FRS solutions in three modes â€” with an SDK, a deployable container, and a plug-and-play (beta) software solution. The software offers four primary services:

Facial recognition

Weapon detection

Space analytics

Live verification to prevent spoofing attempts

performance requirement:

Prerequisites for OpenCV Face Recognition Project:

Microsoft Visual Studio 2019. You'll need Visual Studio C++ for compiling dlib during face-recognition python package installation. ...

Python â€ “3.x (We used python 3.7.10 for this project.

OpenCV â€ “ 4.5. ...

Face-recognition. ...

NumPy â€ “ 1.20

safety requirement:

What are the security risks of facial recognition?

Data Security

Moreover, data breaches involving facial recognition data increase the potential for identity theft, stalking, and harassment because, unlike passwords and credit card information, faces cannot easily be changed. Faces are becoming easier to capture from remote distances and cheaper to collect and store.

Face recognition is a broad problem of identifying or verifying a person in digital images or video frames through the facial biometric pattern and data. The technology collects a set of unique biometric data of each person associated with their face and facial expression to authenticate a person.

Facial recognition systems are a form of mass surveillance that violate the right to privacy and threaten the rights to freedom of peaceful assembly and expression. Facial recognition risks being weaponized by law enforcement against marginalized communities around the world

Facial recognition is a way of recognizing a human face through technology. A facial recognition system uses biometrics to map facial features from a photograph or video. It compares the information with a database of known faces to find a match. Facial recognition can help verify a persona€™s identity, but it also raises privacy issues.

The facial recognition market is expected to grow to $7.7 billion in 2022, an increase from $4 billion in 2017. Theta€™s because facial recognition has many commercial applications. It can be used for everything from surveillance to marketing.

But theta€™s where it gets complicated. If privacy is important to you, you probably want some control over how your personal information â€” your data â€” is used. And here€™s the thing: Your â€œfaceprintâ€ is data.

Performance Profile

We profiled our face recognition system on C64x by running the recognition on a 480x640 image

that contains a single face. The database we used had 10 people. We assumed that the data

that would be processed by the system was available in internal or external memory. Since the

rule based approach we used for decreasing the search space for face detection causes

variability in computation time, we averaged the performance results over a certain number of

input images. The resulting CPU and memory requirements are shown in Table 1. A quick look

at these results reveals that the face detection and eye localization blocks consume most of the

computation time, and the face classification blocks consume most of the memory. These results

are expected since searching for faces and features in still images at multiple scales and

locations is known to be a computationally intensive task, and the classification blocks have to

store the subspace models and the face databases which are large in size. Note that an

increase in database size will linearly increase the CPU and memory requirements of the

classification blocks.

Now, let us discuss these results from the point of view of the face recognition approaches we

have explained in section 2. Our implementation follows the still image processing approach

shown in part (a) of Figure 1. Therefore, based on the results shown in Table 1, we can conclude

that it takes less than 4 seconds to recognize a single face from a still image. Most of this time is

spent during the face detection and eye localization stages. Hence, choosing faster algorithms

for these stages can increase the recognition speed significantly.

Although we did not implement the video-based approach shown in part (b), we can

make some comments about it based on the face normalization and classification blocks we

have implemented. We believe that it is possible to implement a real time face and facial feature

detection and tracking system on C64x. If such as system is available to process the video, the

location of the face and facial features can be known at any time. Then, according to the results

shown in Table 1, normalization and classification takes less than 1 second to complete. This

can make it possible to implement a very fast face recognition system on C64x.

The above comments are all related to the face recognition camera application we have

described in section 3.1. The difference of the face recognition pre-processor application shown

in section 3.2 is to move the classification block to the central server. Large memory

requirements of the classification blocks shown in Table 1 may motivate such an approach.

SPRA874

10 Performance Analysis of Face Recognition Algorithms on TMS320C64x

Finally, if we look at the performance profiles of the two classification algorithms, we notice a

tradeoff between the recognition rate and the computational complexity. Eigenfaces

classification is faster, consumes less memory, and, as we have mentioned in section 4,

provides a lower recognition rate than the segmented linear subspaces method. Similar trade

offs would probably exist for all face classification algorithms.

Table 1. Performance Profile

Number of

Cycles (x 106)

Computation Time

(With a CPU at 500Mhz)

Memory Consumption

for Data

Face detection 1161 2.32 seconds ~ 392 KB

Eye localization 585 1.17 seconds ~ 436 KB

Face normalization 56 0.11 seconds ~ 32 KB

Face classification

(eigenfaces)

18 0.04 seconds ~ 1055 KB

Face classification

(Segmented linear sub.)

22 0.05 seconds ~ 2064 KB

**Chapter -3: System Design**

Introduction

Face recognition systems are part of facial image processing applications and their significance

as a research area are increasing recently. They use biometric information of humans and are

applicable easily instead of fingerprint, iris, signature etc., because these types of biometrics are

not much suitable for non-collaborative people. Face recognition systems are usually applied

and preferred for people and security cameras in metropolitan life. These systems can be used

for crime prevention, video surveillance, person verification, and similar security activities.

Face recognition systems are part of facial image processing applications and their significance as a research area are increasing recently. They use biometric information of humans and are applicable easily instead of fingerprint, iris, signature etc., because these types of biometrics are not much suitable for non-collaborative people. Face recognition systems are usually applied and preferred for people and security cameras in metropolitan life. These systems can be used for crime prevention, video surveillance, person verification, and similar security activities**.** In this we are going to explain our system design.

3.1 Flowcharts

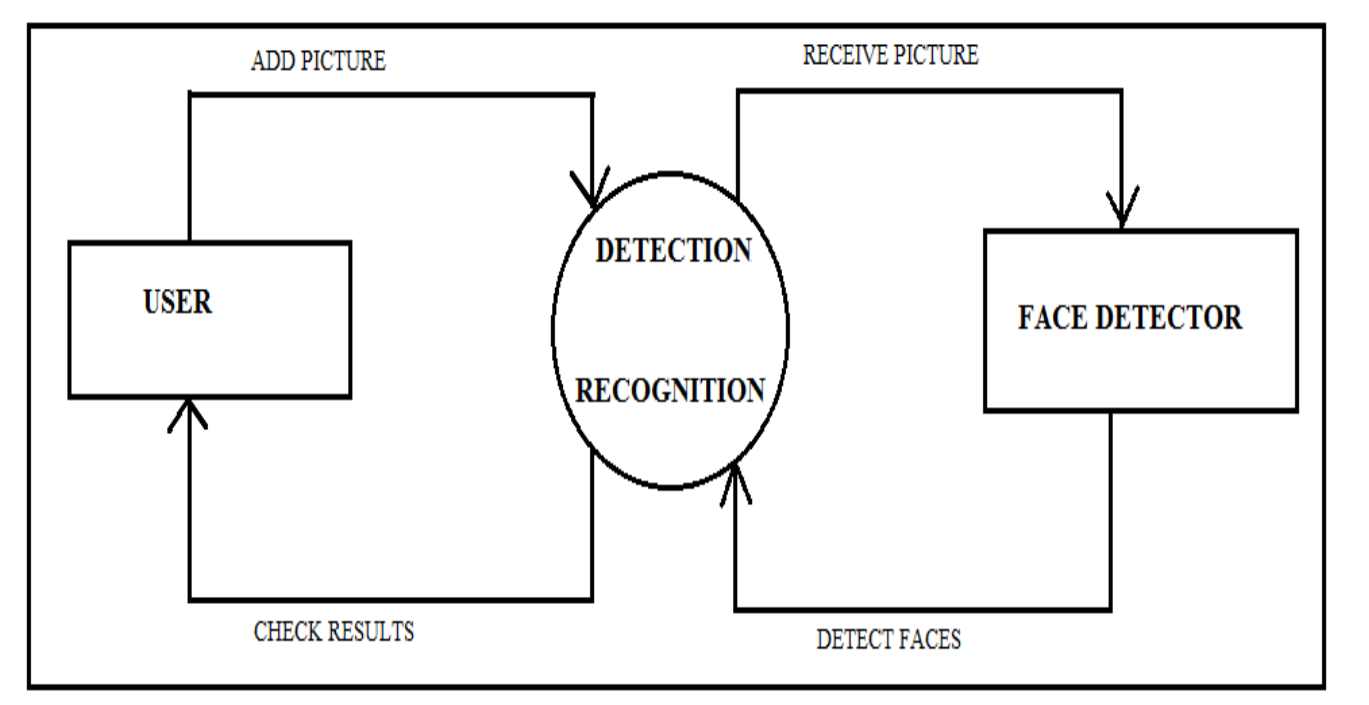


Fig 6

3.2 ER Diagram

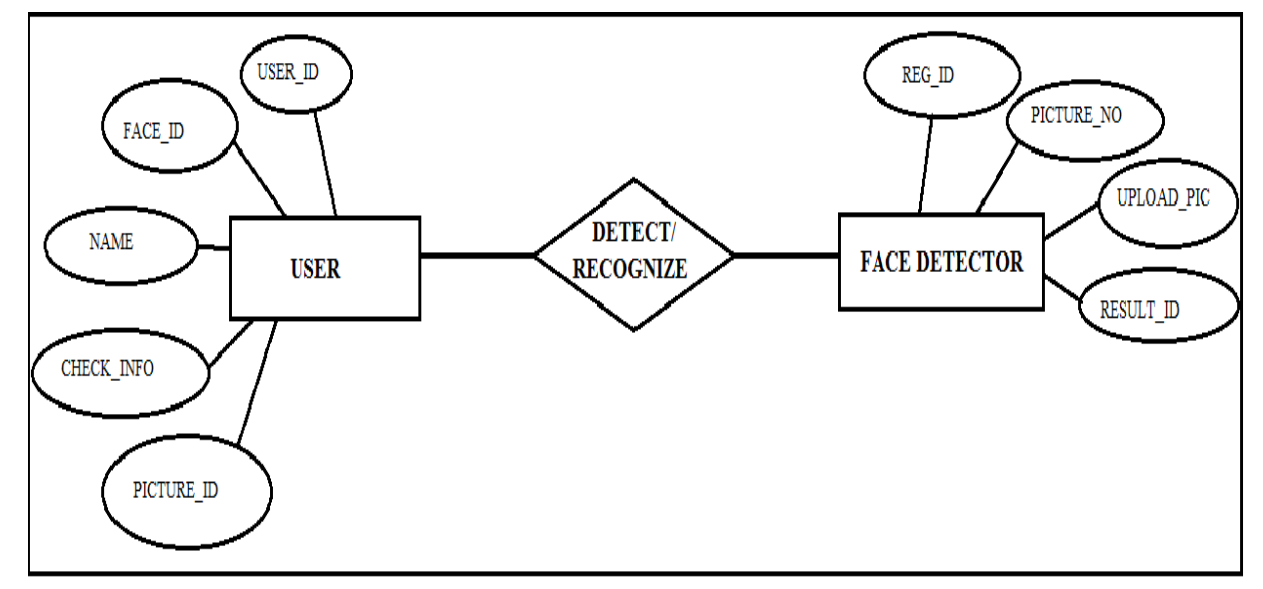


Fig 7

3.3 Design Methodology

A throughout survey has revealed that various methods and combination of these methods can

be applied in development of a new face recognition system. Among the many possible

approaches, we have decided to use a combination of knowledge-based methods for face

detection part and neural network approach for face recognition part. The main reason in this

selection is their smooth applicability and reliability issues. Our face recognition system

approach is given in Figure 2.

A throughout survey has revealed that various methods and combination of these methods can

be applied in development of a new face recognition system. Among the many possible

approaches, we have decided to use a combination of knowledge-based methods for face

detection part and neural network approach for face recognition part. The main reason in this

selection is their smooth applicability and reliability issues. Our face recognition system

approach is given in Figure 2.

A throughout survey has revealed that various methods and combination of these methods can

be applied in development of a new face recognition system. Among the many possible

approaches, we have decided to use a combination of knowledge-based methods for face

detection part and neural network approach for face recognition part. The main reason in this

selection is their smooth applicability and reliability issues. Our face recognition system

approach is given in Figure 2

A throughout survey has revealed that various methods and combination of these methods can

be applied in development of a new face recognition system. Among the many possible

approaches, we have decided to use a combination of knowledge-based methods for face

detection part and neural network approach for face recognition part. The main reason in this

selection is their smooth applicability and reliability issues. Our face recognition system

approach is given in Figure 2

A face recognition system is one of the biometric information processes, its applicability is easier and working range is larger than others, i.e.; fingerprint, iris scanning, signature, etc. A face recognition system is designed, implemented and tested at Atılım University, Mechatronics Engineering Department. The system uses a combination of techniques in two topics; face detection and recognition. The face detection is performed on live acquired images without any application field in mind. Processes utilized in the system are white balance correction, skin like region segmentation, facial feature extraction and face image extraction on a face candidate. Then a face classification method that uses FeedForward Neural Network is integrated in the system. The system is tested with a database generated in the laboratory with 26 people. The tested system has acceptable performance to recognize faces within intended limits. System is also capable of detecting and recognizing multiple faces in live acquired images.

3.3.1 Design of a face recognition system

A throughout survey has revealed that various methods and combination of these methods can

be applied in development of a new face recognition system. Among the many possible

approaches, we have decided to use a combination of knowledge-based methods for face

detection part and neural network approach for face recognition part. The main reason in this

selection is their smooth applicability and reliability issues. Our face recognition system

approach is given in Figure

A throughout survey has revealed that various methods and combination of these methods can

be applied in development of a new face recognition system. Among the many possible

approaches, we have decided to use a combination of knowledge-based methods for face

detection part and neural network approach for face recognition part. The main reason in this

selection is their smooth applicability and reliability issues. Our face recognition system

approach is given in Figure

A throughout survey has revealed that various methods and combination of these methods can

be applied in development of a new face recognition system. Among the many possible

approaches, we have decided to use a combination of knowledge-based methods for face

detection part and neural network approach for face recognition part. The main reason in this

selection is their smooth applicability and reliability issues. Our face recognition system

approach is given in Figure

A throughout survey has revealed that various methods and combination of these methods can be applied in development of a new face recognition system. Among the many possible approaches, we have decided to use a combination of knowledge-based methods for face detection part and neural network approach for face recognition part. The main reason in this selection is their smooth applicability and reliability issues. Our face recognition system approach is given in Figure



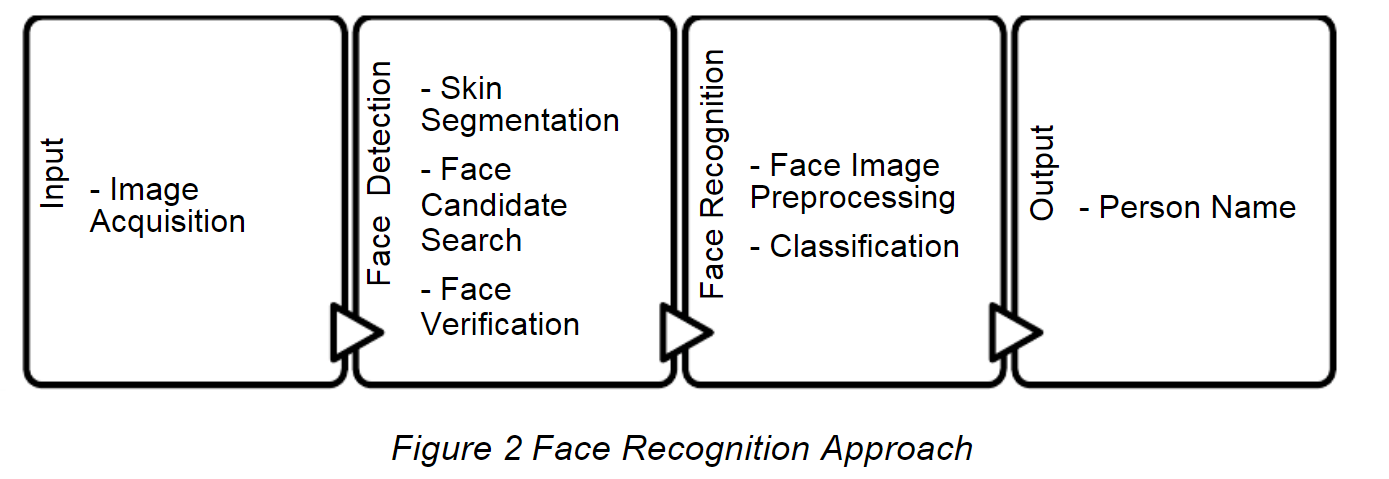


Fig 8

1. Input Part

Input part is prerequisite for face recognition system. Image acquisition operation is performed in

this part. Live captured images are converted to digital data for performing image-processing

computations. These captured images are sent to face detection algorithm.

Input part is prerequisite for face recognition system. Image acquisition operation is performed in this part. Live captured images are converted to digital data for performing image-processing computations. These captured images are sent to face detection algorithm.

**Input part is prerequisite for face recognition system. Image acquisition operation is performed in**

**this part. Live captured images are converted to digital data for performing image-processing**

**computations. These captured images are sent to face detection algorithm.**

2. Face Detection Part

Face detection performs locating and extracting face image operations for face recognition

system. Face detection part algorithm is given in Figure

Face detection performs locating and extracting face image operations for face recognition system. Face detection part algorithm is given in Figure

Face detection performs locating and extracting face image operations for face recognition

system. Face detection part algorithm is given in Figure

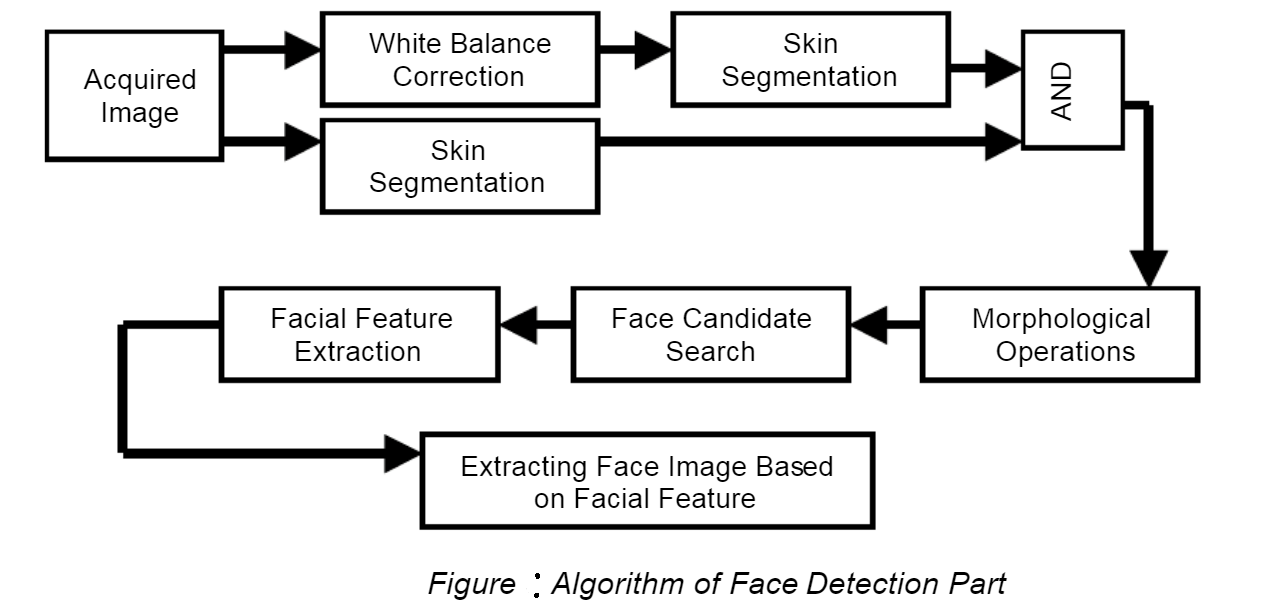


Fig 9

3. Face Recognition Part

Modified face image which is obtained in the Face recognition system, should to be classified to

identify the person in the database. Face recognition part is composed of preprocessing face

image, vectorizing image matrix, database generation, and then classification. The classification

is achieved by using FeedForward Neural Network (FFNN) [13]. Face recognition part algorithm

is given in Figure

Modified face image which is obtained in the Face recognition system, should to be classified to identify the person in the database. Face recognition part is composed of pre-processing face image, vectorizing image matrix, database generation, and then classification. The classification is achieved by using FeedForward Neural Network (FFNN). Face recognition part algorithm is given in Figure



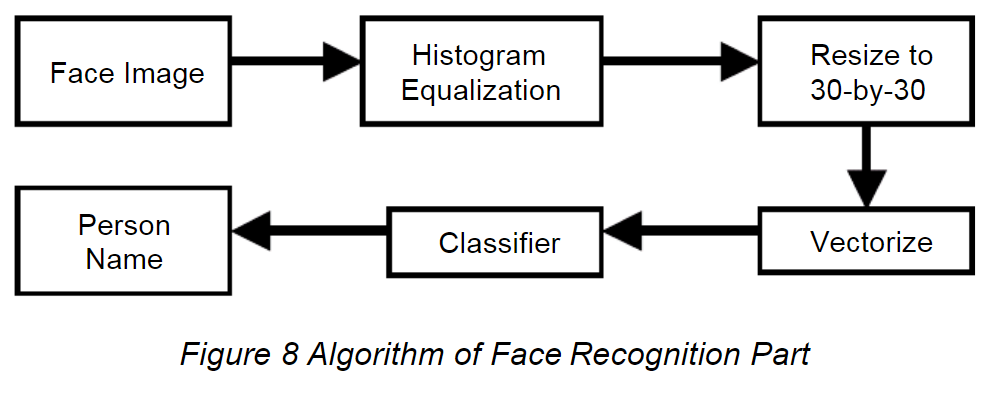


Fig 10

Before classifying the face image, it should be pre-processed. Pre-processing operations are histogram equalizing of grayscale face image, resizing to 30-by-30 pixels, and finally vectorizing the matrix image. Face Image Histogram Equalization Resize to 30-by-30 Vectorize Classifier Person Name

The 15th International Conference on Machine Design and Production June 19 – 22, 2012, Pamukkale, Denizli, Turkey 7 In classifier, FeedForward Neural Network (FFNN) is used . FFNN is the simplest structure in the neural network. This type of network structure is generally used for pattern recognition applications. System network properties are: input layer has 900 inputs, hidden layer has 41 neurons and output layer has 26 neurons. Output layer has 26 neuron since the number of people in database is 26. After structure is generated, then network should be trained to classify the given images with respect to face database. Therefore, face database is created before any tests. A database is created for 26 people with 4 samples for each person. This results 104 training sample. Due to that, 900-by-104 size matrix will be training matrix. Training matrix vector element is arranged with four groups due to the number of samples for each person. Though, first 26 vector element belongs to first samples of 26 people, and it continues. Training matrix’s columns are made from pre-processing image and then vectorizing to face image which generate database. After training matrix and target matrix is created, then training of NN can be performed. Back propagation is used to train the network. Training performance and goal errors are set to 1e-17 to classify given image correctly.

3.4 Software Development Model

With the rapid use of Android OS in mobile devices and related products, face recognition technology is an essential feature, so that mobile devices have a strong personal identity authentication. In this, we propose Android based software development framework for real-time face detection and recognition using OpenCV library, which is applicable in several mobile applications

3.4.1 OpenCV library

In the field of [Artificial Intelligence](https://www.mygreatlearning.com/ai/free-courses/?gl_blog_id=21086), Computer Vision is one of the most interesting and Challenging tasks. Computer Vision acts like a bridge between Computer Software and visualizations around us. It allows computer software to understand and learn about the visualizations in the surroundings. For Example: Based on the colour, shape and size determining the fruit. This task can be very easy for the human brain however in the Computer Vision pipeline, first we gather the data, then we perform the data processing activities and then we train and teach the model to understand how to distinguish between the fruits based on size, shape and colour of fruit. Currently, various packages are present to perform [machine learning](https://www.mygreatlearning.com/pg-program-machine-learning-course/?gl_blog_id=21086), deep learning and computer vision tasks. By far, computer vision is the best module for such complex activities. [OpenCV](https://www.mygreatlearning.com/academy/learn-for-free/courses/computer-vision-for-dummies-with-opencv/?gl_blog_id=21086" \t "_blank) is an open-source library. It is supported by various programming languages such as R, Python. It runs on most of the platforms such as Windows, [Linux](https://www.mygreatlearning.com/academy/learn-for-free/courses/linux-tutorial/?gl_blog_id=21086) and MacOS.

OpenCV is an open source library of image processing algorithms based on C or C++ programming. OpenCV libraries have the advantages such as: Cross-platform, independent of operating system, hardware, and graphics manager; OpenCV is free of charge for non-commercial or commercial applications; fast and easy to use; good scalability including low-level and high-level application development kits for common image or video load, save and capture modules. As the devices with An-droid OS has the capabilities to obtain the benefits provided by OpenCV library which also guarantee real-time, an image pre-processing algorithm is used to form a face recognition system.

**Advantages of OpenCV:**

* OpenCV is an open-source library and is free of cost.
* As compared to other libraries, it is fast since it is written in C/C++.
* It works better on System with lesser RAM
* T supports most of the Operating Systems such as Windows, Linux and MacOS.

**Installation:**

Here we will be focusing on installing OpenCV for python only. We can install OpenCV using pip

1. Using pip:

Using pip, the installation process of OpenCV can be done by using the following command in the command prompt.

**pip install OpenCv-python**

3.4.1 The Raspberry Pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. The Raspberry Pi project originally leaned towards the promotion of teaching basic [computer science](https://en.wikipedia.org/wiki/Computer_science) in schools and in [developing countries](https://en.wikipedia.org/wiki/Developing_countries).

3.4.2 Scikit-image

Scikit-image is an open-source image processing library for the Python programming language. It includes algorithms for segmentation, geometric transformations, color space manipulation, analysis, filtering, morphology, feature detection, and more. It is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy" \o "SciPy).

3.5 Database Design

When benchmarking an algorithm it is recommendable to use a standard test data set for researchers to be able to directly compare the results. While there are many databases in use currently, the choice of an appropriate database to be used should be made based on the task given (aging, expressions, lighting etc). Another way is to choose the data set specific to the property to be tested (e.g. how algorithm behaves when given images with lighting changes or images with different facial expressions).

3.5.1 ER Diagram

3.5.2 Schema

3.6 DFD’s (Data Flow Diagrams)

The  **DFD** is a diagram used to show the overall data management of the project. It has 3 main levels that shows the Face Recognition System data handling which is the DFD Level 0, 1, and 2. These DFD levels illustrates the Face Recognition data management concept from the basics up to specific details. The DFD Level 0, 1 and 2 has their part in explaining the data handling of Face Recognition System. The DFD Level 0 (context Diagram) shows the overall function of Face Recognition System in a single process. It also serves as the main basis as you make the proceeding Levels. DFD Level 1 broadens that DFD Level 0 and the DFD Level 2 includes the databases of the all the data that will be part of the Project’s main function.

**Face Recognition System DFD Level 0**

The DFD Diagram level 0 shows the abstract view of Face Recognition System and represented as a single process with external entities and main data. This level introduces the main function of Face Recognition System in general that is why it is called as the context diagram.

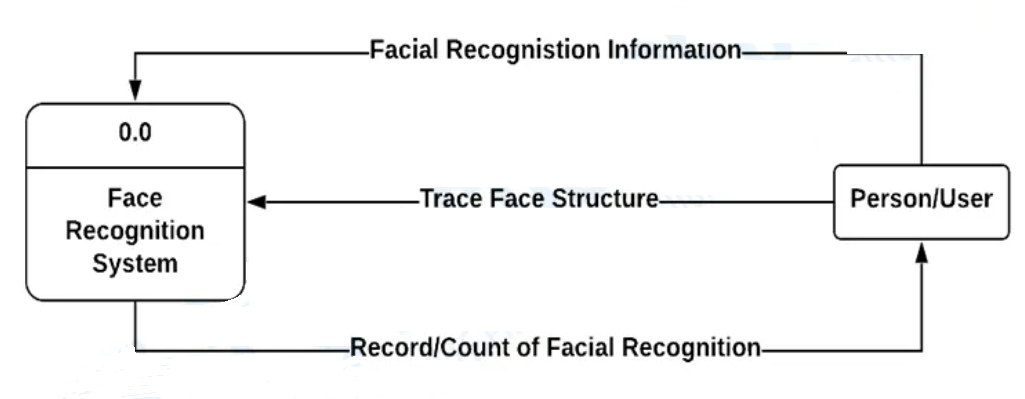
****

Fig 11

**Face Recognition System DFD Level 1**

Next to the context diagram is the level 1 data flow diagram. The content of DFD level 1 is the broadened idea form the DFD level 0. It reveals further processing information as well the data and processes that completes the Face Recognition System function.

This level provides a broad overview and greater depth of the context diagram. The single process node from the context diagram is broken down into sub processes and includes all the needed data to complete the whole concept. It lists all of the major sub-processes that make up the entire system and its relationships to external entities are depicted in Level 0. DFD level 1 can be thought of as a “detonated view” of the context diagram.

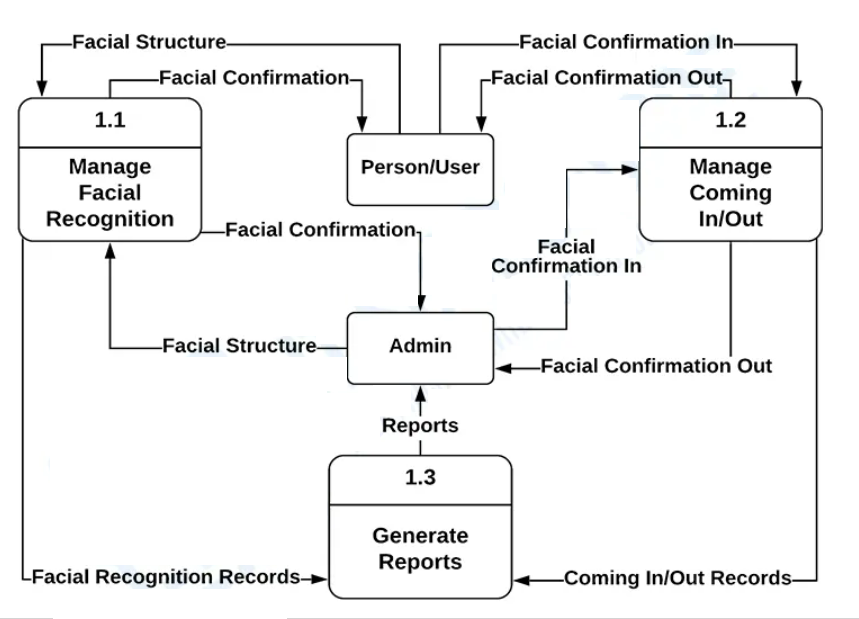
****



Fig 12

**Face Recognition System DFD Level 2**

After presenting the Face Recognition System DFD levels 0 and 1, last is level 2. Considered as the highest abstraction level, DFD Level 2 is the detailed version of DFD level 1. The DFD Level 2 represent the database or data storage used to secure all the data that moves inside the System.

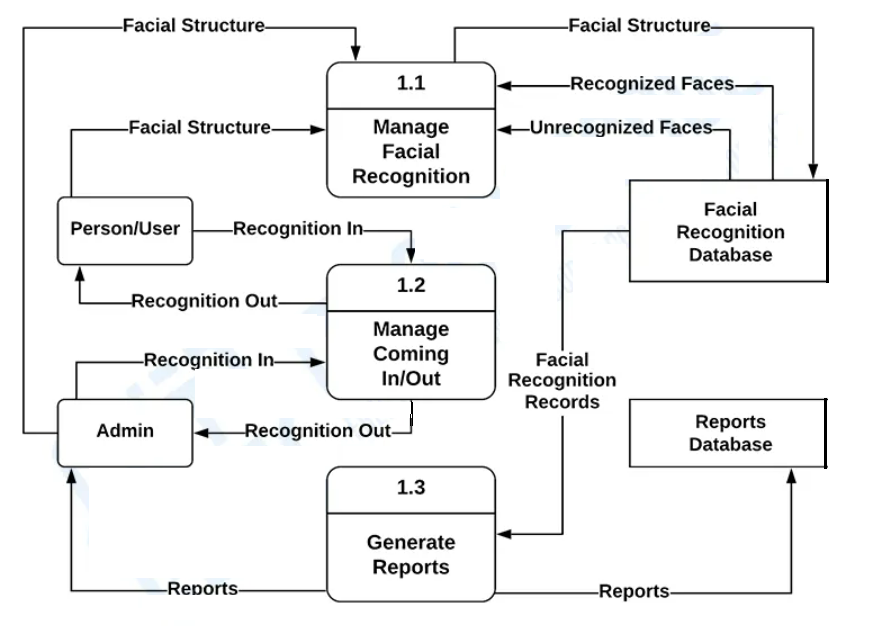
****

Fig 13

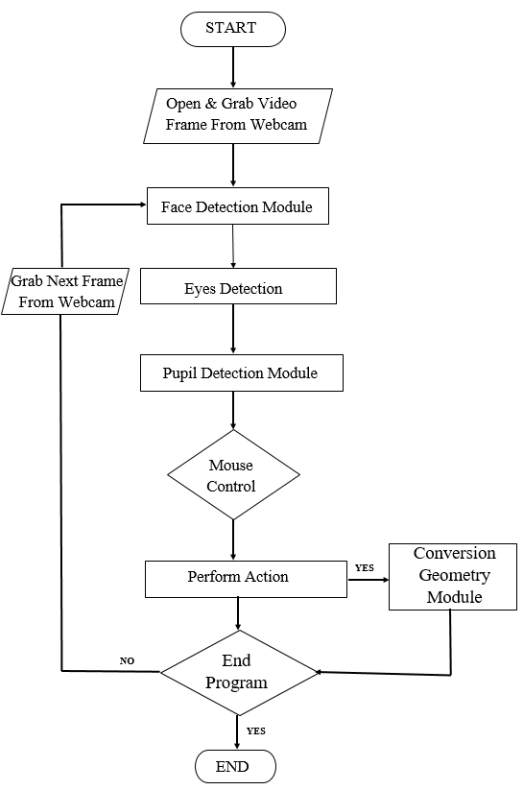
3.7 Activity Diagram****

Fig 14

3.8 Class Diagram

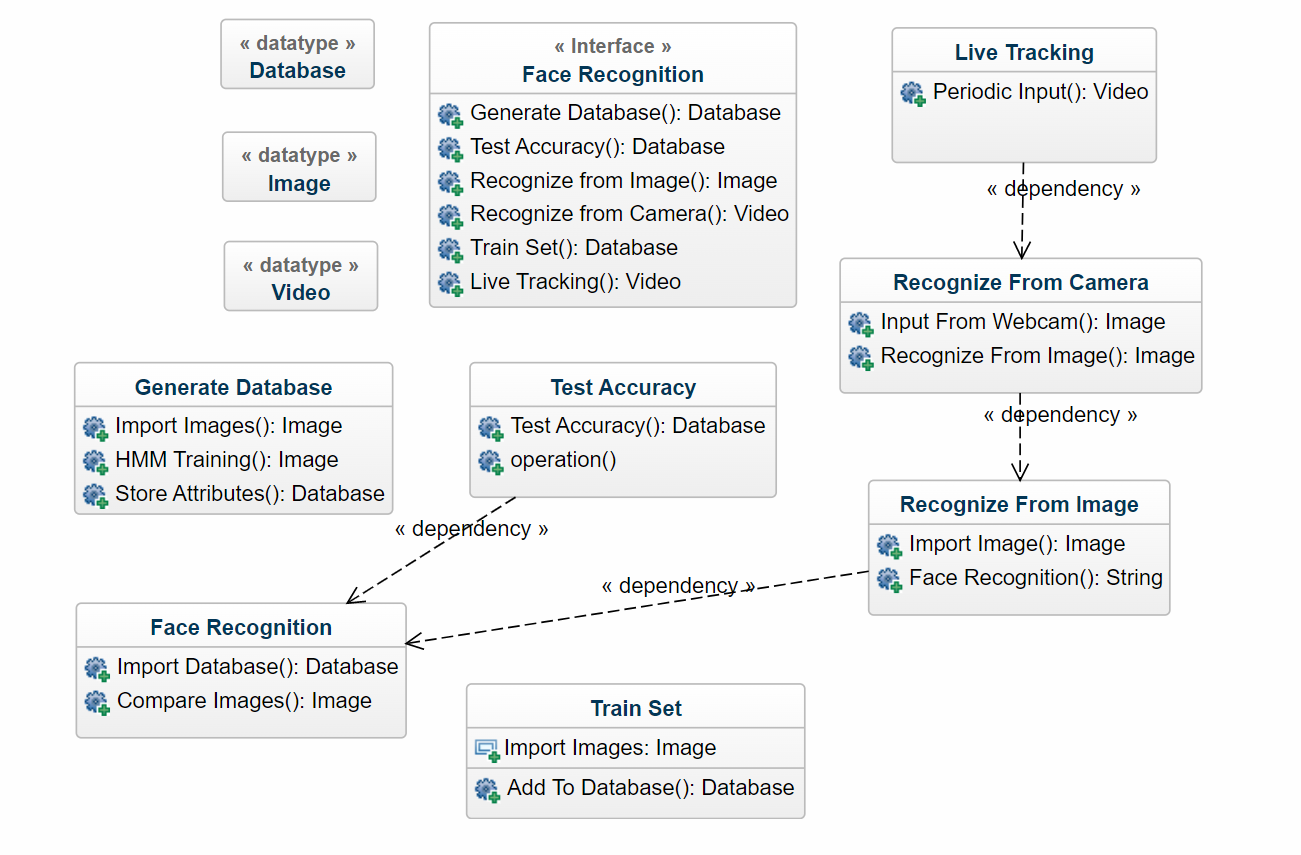
****

Fig 15

3.9 Sequence Diagram

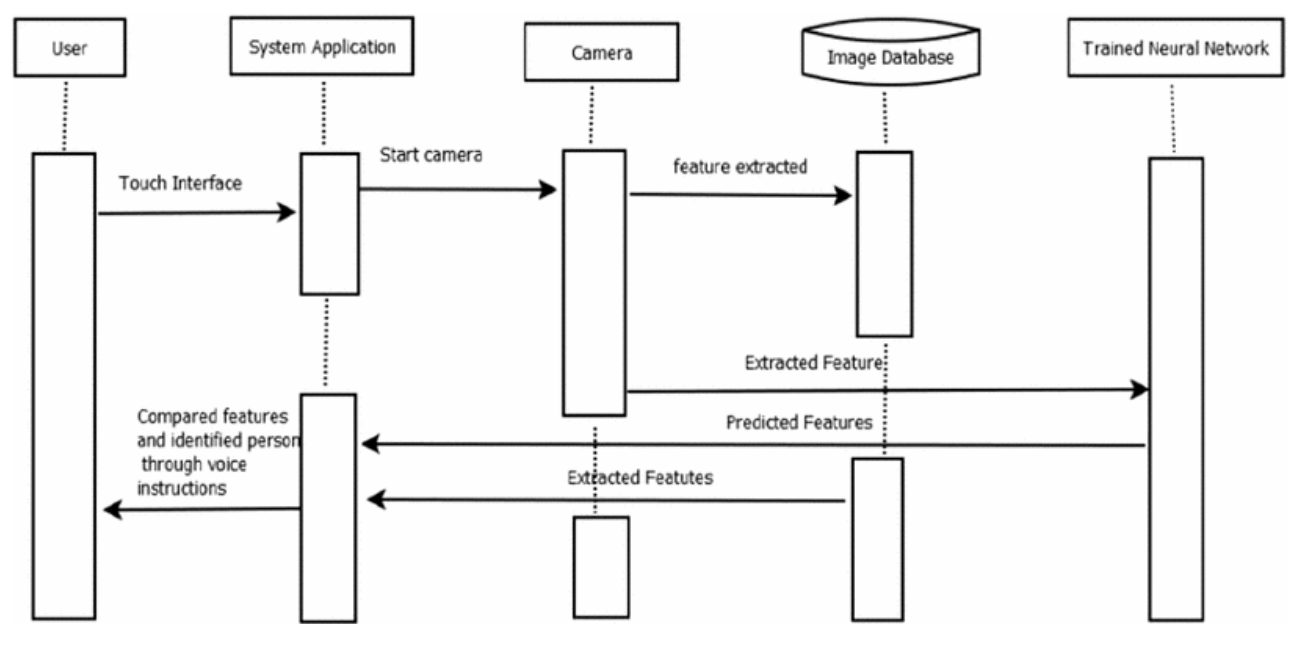
****

Fig 16

**Chapter 4 - System Implementation**

4.1 Coding

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

    <title>FDR</title>

    <link rel="icon" type="image/x-icon" href="C:\Users\ARIF ALAM ANSARI\Downloads\favicon2.jpg">

</head>

<body>

    <header class="text-gray-600 body-font">

        <div class="container mx-auto flex flex-wrap p-5 flex-col md:flex-row items-center">

          <a class="flex title-font font-medium items-center text-gray-900 mb-4 md:mb-0">

            <svg xmlns="http://www.w3.org/2000/svg" fill="none" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-10 h-10 text-white p-2 bg-indigo-500 rounded-full" viewBox="0 0 24 24">

              <!--<path d="M12 2L2 7l10 5 10-5-10-5zM2 17l10 5 10-5M2 12l10 5 10-5"></path> -->

            </svg>

            <span class="ml-3 text-xl">Face Detection</span>

          </a>

          <nav class="md:mr-auto md:ml-4 md:py-1 md:pl-4 md:border-l md:border-gray-400 flex flex-wrap items-center text-base justify-center">

              <a href="/"class="mr-5 hover:text-gray-900">Home </a>

              <a href="/"class="mr-5 hover:text-gray-900">About</a>

            <a href="/"class="mr-5 hover:text-gray-900">Services</a>

            <a href="/"class="mr-5 hover:text-gray-900">Contact</a>

          </nav>

          <button class="inline-flex items-center bg-gray-100 border-0 py-1 px-3 focus:outline-none hover:bg-gray-200 rounded text-base mt-4 md:mt-0"><a href="file:///C:/Users/ARIF%20ALAM%20ANSARI/Desktop/minipro/2nd.html">Check our product</a>

            <svg fill="none" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-4 h-4 ml-1" viewBox="0 0 24 24">

              <path d="M5 12h14M12 5l7 7-7 7"></path>

            </svg>

          </button>

        </div>

      </header>

      <section class="text-gray-600 body-font">

        <div class="container mx-auto flex px-5 py-24 md:flex-row flex-col items-center">

          <div class="lg:flex-grow md:w-1/2 lg:pr-24 md:pr-16 flex flex-col md:items-start md:text-left mb-16 md:mb-0 items-center text-center">

            <h1 class="title-font sm:text-4xl text-3xl mb-4 font-medium text-gray-900">Face Detection and

              <br class="hidden lg:inline-block">Recognition

            </h1>

            <p class="mb-8 leading-relaxed">Face detection and recognition, a new technology for taking attendance, identifying person, and traffic control system.</p>

            <div class="flex justify-center">

              <button class="inline-flex text-white bg-indigo-500 border-0 py-2 px-6 focus:outline-none hover:bg-indigo-600 rounded text-lg">Face scan</button>

              <button class="ml-4 inline-flex text-gray-700 bg-gray-100 border-0 py-2 px-6 focus:outline-none hover:bg-gray-200 rounded text-lg">Details</button>

            </div>

          </div>

          <div class="lg:max-w-lg lg:w-full md:w-1/2 w-5/6">

            <img class="object-cover object-center rounded" alt="hero" src="  https://miro.medium.com/max/1320/1\*enzZrRQ\_EwtfJJKOffrcFg.png">

          </div>

        </div>

      </section>

      <hr>

      <section class="text-gray-600 body-font">

        <div class="container px-5 py-24 mx-auto">

          <div class="flex flex-wrap -m-4 text-center">

            <div class="p-4 sm:w-1/4 w-1/2">

              <h2 class="title-font font-medium sm:text-4xl text-3xl text-gray-900">2.7K</h2>

              <p class="leading-relaxed">Users</p>

            </div>

            <div class="p-4 sm:w-1/4 w-1/2">

              <h2 class="title-font font-medium sm:text-4xl text-3xl text-gray-900">1.8K</h2>

              <p class="leading-relaxed">Subscribes</p>

            </div>

            <div class="p-4 sm:w-1/4 w-1/2">

              <h2 class="title-font font-medium sm:text-4xl text-3xl text-gray-900">35</h2>

              <p class="leading-relaxed">Downloads</p>

            </div>

            <div class="p-4 sm:w-1/4 w-1/2">

              <h2 class="title-font font-medium sm:text-4xl text-3xl text-gray-900">4</h2>

              <p class="leading-relaxed">Products</p>

            </div>

          </div>

        </div>

      </section>

      <hr>

      <section class="text-gray-600 body-font relative">

        <div class="container px-5 py-24 mx-auto">

          <div class="flex flex-col text-center w-full mb-12">

            <h1 class="sm:text-3xl text-2xl font-medium title-font mb-4 text-gray-900">Contact Us</h1>

            <p class="lg:w-2/3 mx-auto leading-relaxed text-base">Fill the required details</p>

          </div>

          <div class="lg:w-1/2 md:w-2/3 mx-auto">

            <div class="flex flex-wrap -m-2">

              <div class="p-2 w-1/2">

                <div class="relative">

                  <label for="name" class="leading-7 text-sm text-gray-600">Name</label>

                  <input type="text" id="name" name="name" class="w-full bg-gray-100 bg-opacity-50 rounded border border-gray-300 focus:border-indigo-500 focus:bg-white focus:ring-2 focus:ring-indigo-200 text-base outline-none text-gray-700 py-1 px-3 leading-8 transition-colors duration-200 ease-in-out">

                </div>

              </div>

              <div class="p-2 w-1/2">

                <div class="relative">

                  <label for="email" class="leading-7 text-sm text-gray-600">Email</label>

                  <input type="email" id="email" name="email" class="w-full bg-gray-100 bg-opacity-50 rounded border border-gray-300 focus:border-indigo-500 focus:bg-white focus:ring-2 focus:ring-indigo-200 text-base outline-none text-gray-700 py-1 px-3 leading-8 transition-colors duration-200 ease-in-out">

                </div>

              </div>

              <div class="p-2 w-full">

                <div class="relative">

                  <label for="message" class="leading-7 text-sm text-gray-600">Message</label>

                  <textarea id="message" name="message" class="w-full bg-gray-100 bg-opacity-50 rounded border border-gray-300 focus:border-indigo-500 focus:bg-white focus:ring-2 focus:ring-indigo-200 h-32 text-base outline-none text-gray-700 py-1 px-3 resize-none leading-6 transition-colors duration-200 ease-in-out"></textarea>

                </div>

              </div>

              <div class="p-2 w-full">

                <button class="flex mx-auto text-white bg-indigo-500 border-0 py-2 px-8 focus:outline-none hover:bg-indigo-600 rounded text-lg">Submit</button>

              </div>

              <div class="p-2 w-full pt-8 mt-8 border-t border-gray-200 text-center">

                <a class="text-indigo-500">example@email.com</a>

                <p class="leading-normal my-5">Arif, Adiba, Amritanshu, Aditya

                  <br>NIET GR. NOIDA

                </p>

                <span class="inline-flex">

                  <a class="text-gray-500">

                    <svg fill="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-5 h-5" viewBox="0 0 24 24">

                      <path d="M18 2h-3a5 5 0 00-5 5v3H7v4h3v8h4v-8h3l1-4h-4V7a1 1 0 011-1h3z"></path>

                    </svg>

                  </a>

                  <a class="ml-4 text-gray-500">

                    <svg fill="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-5 h-5" viewBox="0 0 24 24">

                      <path d="M23 3a10.9 10.9 0 01-3.14 1.53 4.48 4.48 0 00-7.86 3v1A10.66 10.66 0 013 4s-4 9 5 13a11.64 11.64 0 01-7 2c9 5 20 0 20-11.5a4.5 4.5 0 00-.08-.83A7.72 7.72 0 0023 3z"></path>

                    </svg>

                  </a>

                  <a class="ml-4 text-gray-500">

                    <svg fill="none" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-5 h-5" viewBox="0 0 24 24">

                      <rect width="20" height="20" x="2" y="2" rx="5" ry="5"></rect>

                      <path d="M16 11.37A4 4 0 1112.63 8 4 4 0 0116 11.37zm1.5-4.87h.01"></path>

                    </svg>

                  </a>

                  <a class="ml-4 text-gray-500">

                    <svg fill="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-5 h-5" viewBox="0 0 24 24">

                      <path d="M21 11.5a8.38 8.38 0 01-.9 3.8 8.5 8.5 0 01-7.6 4.7 8.38 8.38 0 01-3.8-.9L3 21l1.9-5.7a8.38 8.38 0 01-.9-3.8 8.5 8.5 0 014.7-7.6 8.38 8.38 0 013.8-.9h.5a8.48 8.48 0 018 8v.5z"></path>

                    </svg>

                  </a>

                </span>

              </div>

            </div>

          </div>

        </div>

      </section>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <link rel="icon" type="image/x-icon" href="C:\Users\ARIF ALAM ANSARI\Downloads\favicon2.jpg">

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

    <title>FDR</title>

</head>

<body>

    <header class="text-gray-600 body-font">

        <div class="container mx-auto flex flex-wrap p-5 flex-col md:flex-row items-center">

          <a class="flex title-font font-medium items-center text-gray-900 mb-4 md:mb-0">

            <svg xmlns="http://www.w3.org/2000/svg" fill="none" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-10 h-10 text-white p-2 bg-indigo-500 rounded-full" viewBox="0 0 24 24">

             <!-- <path d="M12 2L2 7l10 5 10-5-10-5zM2 17l10 5 10-5M2 12l10 5 10-5"></path> -->

            </svg>

            <span class="ml-3 text-xl">FDR</span>

          </a>

          <nav class="md:mr-auto md:ml-4 md:py-1 md:pl-4 md:border-l md:border-gray-400 flex flex-wrap items-center text-base justify-center">

            <a class="mr-5 hover:text-gray-900" href="file:///C:/Users/ARIF%20ALAM%20ANSARI/Desktop/minipro/index.html">Home</a>

            <a class="mr-5 hover:text-gray-900">About </a>

            <a class="mr-5 hover:text-gray-900">Service</a>

            <a class="mr-5 hover:text-gray-900">Details</a>

          </nav>

        </div>

      </header>

      <section class="text-gray-600 body-font">

        <div class="container px-5 py-24 mx-auto">

          <div class="flex flex-wrap w-full mb-20">

            <div class="lg:w-1/2 w-full mb-6 lg:mb-0">

              <h1 class="sm:text-3xl text-2xl font-medium title-font mb-2 text-gray-900">Products varient</h1>

              <div class="h-1 w-20 bg-indigo-500 rounded"></div>

            </div>

            <p class="lg:w-1/2 w-full leading-relaxed text-gray-500">Facial recognition is a way of identifying or confirming an individual's identity using their face. Facial recognition systems can be used to identify people in photos, videos, or in real-time. Facial recognition is a category of biometric security.</p>

          </div>

          <div class="flex flex-wrap -m-4">

            <div class="xl:w-1/4 md:w-1/2 p-4">

              <div class="bg-gray-100 p-6 rounded-lg">

                <img class="h-40 rounded w-full object-cover object-center mb-6" src="https://www.kumc.edu/images/news/20\_zoom\_surgery\_720x400.jpg" alt="content">

                <h3 class="tracking-widest text-indigo-500 text-xs font-medium title-font">Product 1</h3>

                <h2 class="text-lg text-gray-900 font-medium title-font mb-4">Attendance</h2>

                <p class="leading-relaxed text-base">The system works on face recognition where each student in the class is photographed and their details are stored in a server.</p>

              </div>

            </div>

            <div class="xl:w-1/4 md:w-1/2 p-4">

              <div class="bg-gray-100 p-6 rounded-lg">

                <img class="h-40 rounded w-full object-cover object-center mb-6" src="  https://miro.medium.com/max/1320/1\*enzZrRQ\_EwtfJJKOffrcFg.png" alt="content">

                <h3 class="tracking-widest text-indigo-500 text-xs font-medium title-font">Product 2</h3>

                <h2 class="text-lg text-gray-900 font-medium title-font mb-4">Live face Recognition</h2>

                <p class="leading-relaxed text-base">Facial recognition is a way of identifying or confirming an individual's identity using their face.</p>

              </div>

            </div>

            <div class="xl:w-1/4 md:w-1/2 p-4">

              <div class="bg-gray-100 p-6 rounded-lg">

                <img class="h-40 rounded w-full object-cover object-center mb-6" src="  https://thelogicalindian.com/h-upload/2020/01/27/158723-facefb.jpg" alt="content">

                <h3 class="tracking-widest text-indigo-500 text-xs font-medium title-font">product 3</h3>

                <h2 class="text-lg text-gray-900 font-medium title-font mb-4">Traffic analyser</h2>

                <p class="leading-relaxed text-base">Facial recognition is a way of identifying or confirming an individual's identity using their face.</p>

              </div>

            </div>

            <div class="xl:w-1/4 md:w-1/2 p-4">

              <div class="bg-gray-100 p-6 rounded-lg">

                <img class="h-40 rounded w-full object-cover object-center mb-6" src="  https://wordpress.electronicid.eu/wp-content/uploads/2020/05/IDentity-verification-1.png" alt="content">

                <h3 class="tracking-widest text-indigo-500 text-xs font-medium title-font">Product 4</h3>

                <h2 class="text-lg text-gray-900 font-medium title-font mb-4">Online id varifiction</h2>

                <p class="leading-relaxed text-base">A facial recognition system uses biometrics to map facial features from a photograph or video.</p>

              </div>

            </div>

          </div>

        </div>

      </section>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

    <title>FDR</title>

    <link rel="icon" type="image/x-icon" href="C:\Users\ARIF ALAM ANSARI\Downloads\favicon2.jpg">

</head>

<body>

    <header class="text-gray-600 body-font">

        <div class="container mx-auto flex flex-wrap p-5 flex-col md:flex-row items-center">

          <a class="flex title-font font-medium items-center text-gray-900 mb-4 md:mb-0">

            <svg xmlns="http://www.w3.org/2000/svg" fill="none" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-10 h-10 text-white p-2 bg-indigo-500 rounded-full" viewBox="0 0 24 24">

              <!--<path d="M12 2L2 7l10 5 10-5-10-5zM2 17l10 5 10-5M2 12l10 5 10-5"></path> -->

            </svg>

            <span class="ml-3 text-xl">Face Detection</span>

          </a>

          <nav class="md:mr-auto md:ml-4 md:py-1 md:pl-4 md:border-l md:border-gray-400 flex flex-wrap items-center text-base justify-center">

            <a class="mr-5 hover:text-gray-900" href="file:///C:/Users/ARIF%20ALAM%20ANSARI/Desktop/minipro/index.html">Home</a>

              <a href="/"class="mr-5 hover:text-gray-900">About</a>

            <a href="/"class="mr-5 hover:text-gray-900">Services</a>

            <a href="/"class="mr-5 hover:text-gray-900">Contact</a>

          </nav>

        </div>

      </header>

      <section class="text-gray-600 body-font relative">

        <div class="container px-5 py-24 mx-auto flex sm:flex-nowrap flex-wrap">

          <div class="lg:w-2/3 md:w-1/2 bg-gray-300 rounded-lg overflow-hidden sm:mr-10 p-10 flex items-end justify-start relative">

            <iframe width="100%" height="100%" class="absolute inset-0" frameborder="0" title="map" marginheight="0" marginwidth="0" scrolling="no" src="https://maps.google.com/maps?width=100%&height=600&hl=en&q=%C4%B0zmir+(My%20Business%20Name)&ie=UTF8&t=&z=14&iwloc=B&output=embed" style="filter: grayscale(1) contrast(1.2) opacity(0.4);"></iframe>

            <div class="bg-white relative flex flex-wrap py-6 rounded shadow-md">

              <div class="lg:w-1/2 px-6">

                <h2 class="title-font font-semibold text-gray-900 tracking-widest text-xs">ADDRESS</h2>

                <p class="mt-1">Photo booth tattooed prism, portland taiyaki hoodie neutra typewriter</p>

              </div>

              <div class="lg:w-1/2 px-6 mt-4 lg:mt-0">

                <h2 class="title-font font-semibold text-gray-900 tracking-widest text-xs">EMAIL</h2>

                <a class="text-indigo-500 leading-relaxed">example@email.com</a>

                <h2 class="title-font font-semibold text-gray-900 tracking-widest text-xs mt-4">PHONE</h2>

                <p class="leading-relaxed">123-456-7890</p>

              </div>

            </div>

          </div>

          <div class="lg:w-1/3 md:w-1/2 bg-white flex flex-col md:ml-auto w-full md:py-8 mt-8 md:mt-0">

            <h2 class="text-gray-900 text-lg mb-1 font-medium title-font">Feedback</h2>

            <p class="leading-relaxed mb-5 text-gray-600">Post-ironic portland shabby chic echo park, banjo fashion axe</p>

            <div class="relative mb-4">

              <label for="name" class="leading-7 text-sm text-gray-600">Name</label>

              <input type="text" id="name" name="name" class="w-full bg-white rounded border border-gray-300 focus:border-indigo-500 focus:ring-2 focus:ring-indigo-200 text-base outline-none text-gray-700 py-1 px-3 leading-8 transition-colors duration-200 ease-in-out">

            </div>

            <div class="relative mb-4">

              <label for="email" class="leading-7 text-sm text-gray-600">Email</label>

              <input type="email" id="email" name="email" class="w-full bg-white rounded border border-gray-300 focus:border-indigo-500 focus:ring-2 focus:ring-indigo-200 text-base outline-none text-gray-700 py-1 px-3 leading-8 transition-colors duration-200 ease-in-out">

            </div>

            <div class="relative mb-4">

              <label for="message" class="leading-7 text-sm text-gray-600">Message</label>

              <textarea id="message" name="message" class="w-full bg-white rounded border border-gray-300 focus:border-indigo-500 focus:ring-2 focus:ring-indigo-200 h-32 text-base outline-none text-gray-700 py-1 px-3 resize-none leading-6 transition-colors duration-200 ease-in-out"></textarea>

            </div>

            <button class="text-white bg-indigo-500 border-0 py-2 px-6 focus:outline-none hover:bg-indigo-600 rounded text-lg">Click Here</button>

            <p class="text-xs text-gray-500 mt-3">Chicharrones blog helvetica normcore iceland tousled brook viral artisan.</p>

          </div>

        </div>

      </section>

</body>

</html>

4.2 Testing

For testing we have used SDLC. The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. A focus on the SDLC (software development life cycle) can ensure high quality, better functioning software. The **SDLC** helps schedule planning, building, testing, and maintaining an application. However, it is hard to implement the **software development life cycle** wisely and adequately without a good understanding of its phases. In this article, you will read about the detailed stages of the SDLC and how they can be used to provide customers with high-quality products.

The testing stage is completed before releasing the product to users. Most tests (if not all) should be automated, especially if you have implemented pipeline. The goal of the testing phase is to ensure that every feature works as expected. How can you clarify the testing process and choose the right types of test for your project? You will find the answers to these and other questions in our article about software testing and its importance for the SDLC process.

This is the main testing phase in the SDLC, as the project is divided in small modules in the previous phase then the modules will be integrated together to test the system as whole. This is to make sure that the modules work together as intended by the developer (as in the specifications) and required by users. It also checks for bugs, errors and ensure the system is able to work in the intended platform. After ensuring that the product had solved the problem the system is then delivered to the customers.

Diagram

Description automatically generated

Fig 17

A systems development life cycle is composed of a number of clearly defined and distinct work phases which are used by systems engineers and systems developers to plan for, design, build, test, and deliver information system. Like anything that is manufactured on an assembly line, an SDLC aims to produce high-quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly defined phase, within scheduled time frames and cost estimates. Computer systems are complex and often (especially with the recent rise of service oriented architecture) link multiple traditional systems potentially supplied by different software vendors. To manage this level of complexity, a number of SDLC models or methodologies have been created, such as waterfall, spiral, Agil software development, rapid prototyping, incremental, and synchronize and stabilize.

SDLC can be described along a spectrum of agile to iterative to sequential methodologies. Agile methodologies, such as XP and Scrum focus on lightweight processes which allow for rapid changes (without necessarily following the pattern of SDLC approach) along the development cycle, iterative methodologies, such as Rational unified process and dynamic system method, focus on limited project scope and expanding or improving products by multiple iterations. Sequential or big-design-up-front (BDUF) models, such as waterfall, focus on complete and correct planning to guide large projects and risks to successful and predictable results. Other models, such as anamorphic development, tend to focus on a form of development that is guided by project scope and adaptive iterations of feature development.

In project management a project can be defined both with a project life cycle (PLC) and an SDLC, during which slightly different activities occur. According to Taylor (2004), "the project life cycle encompasses all the activities of the project, while the systems development life cycle focuses on realizing the product requirement.

The SDLC is not a methodology per se, but rather a description of the phases in the life cycle of a software application. In a broad sense, these phases are investigation, analysis, design, build, test, implement, and maintenance and support. All software development methodologies follow the SDLC phases but the method of doing that varies vastly between methodologies. In the Scrum framework, for example, one could say a single user story goes through all the phases of the SDLC within a single two-week sprint. Contrast this to the waterfall methodology, as another example, where every business requirement (recorded in the analysis phase of the SDLC in a document called the Business Requirements Specification) is translated into feature/functional descriptions (recorded in the design phase in a document called the Functional Specification) which are then all built in one go as a collection of solution features typically over a period of three to nine months, or more] These methodologies are different approaches, yet they both contain the SDLC phases in which a requirement is born, then travels through the life cycle phases ending in the final phase of maintenance and support, after-which the whole life cycle typically starts again for a subsequent version of the software application.

4.3 Snapshots

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application, Teams

Description automatically generated**

**Graphical user interface, website

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Chapter 5- Conclusions and Future Scope**

5.1 Conclusion

Since computer vision is a hot issue these days, systems like Human Activity Recognition systems are extremely helpful and effective for a wide range of applications, including surveillance and monitoring, as well as assisting the elderly and blind. This not only provides added comfort to end-users, but it may also be applied into other organisations to lessen employee effort. The model performs admirably on video streams and mediocrely on picture data. Because of the convenience and difficulties that the system gives and addresses, activity recognition systems are quite important in today's world. The demand for activity identification for monitoring and surveillance, video segmentation, and other applications is increasing, and this technology may substantially assist. This technology serves as a foundation for a variety of additional activity recognition applications. As a result, for general or specialised objectives, this method is extremely advantageous for both individuals and companies.

* Data selection: suitable data to capture the action may help to improve performance of action recognition.
* Approach of recognition: deep learning-based methods achieved superior performance.
* Multiple-modal: current research highlighted that multi-modal fusion can efficiently improve the performance.

While taking into account the wide range of body positions, background noise, and camera movement, recognizing human activity is still a challenge a common issue in machine vision or with the comprehension of video data in order to locate activities in a the first is a specific input, such as photographs and videos. The next step is to choose relevant data to record. the activity in addition, there is a substantial algorithm. It must be utilised to recognise human behaviour. Understanding and communication are important aspects of human action techniques based on deep neural networks, extraction have a strong performance Aside from that, main and individual activities are classified the identification of human-to-human interactions The detection of objects and activities has gotten more sophisticated the newest hot issues in science

5.2 Future Scope

This project has a lot of potential in the future. To begin, video identification code may be fine-tuned via transfer learning, and much larger datasets can be employed to improve the model's accuracy. Furthermore, web and mobile apps can be developed that call these python scripts via an API call to provide activity recognition on users' mobile devices, as well as assist the elderly and blind in understanding and interacting with their surroundings, and many other real-time activity recognition applications.

Using online training, the posture classifier system's results may be improved over time. One of the most difficult components of training a posture classification system is feature selection. The most popular way is offline: there is a collection of both excellent and terrible examples. developed, and the most descriptive characteristics are selected This, however, is not always the case. We end up choosing examples that impede the training process from grasping the core of the problem. specifying a class's features This is because we usually do not know what we are looking for. These are some excellent examples to utilise.

As a result, it would be ideal if we could devise a method that we could use. As a result, it would be ideal if we could devise a system that would allow us to automatically collect these samples and grow our training set. The findings of the posture categorization system can be improved over time through online training. Feature selection is one of the most difficult aspects of training a posture classification system. The most common method is offline: there are several instances, both good and bad. generated, and the most descriptive traits are chosen However, this is not always the case. We end up selecting instances that prevent the training process from correctly capturing the core of the issue. the distinguishing characteristics of a class This is because we frequently have no notion what we're looking for. These are some excellent examples to utilise. It would be fantastic if we could create a system that could do this. We'll be able to pull it off. When the model is pruned and the classifier encounters a comparable example again, it will properly categorise it. The classifier's accuracy will then steadily increase, resulting in more robust action recognition.

Automated posture selection would be an excellent addition to the system presented in this thesis, since it would allow for top-down construction of a variety of activities. Pose selection is currently done by hand, which frequently leads to the selection of too similar pose categories or the removal of a vital posture category. Hierarchical The group tree structure is a good starting point for the system. Because the tree provides a similarity score, postures would be chosen automatically. The next step would be to feed the system, which may be done in a variety of ways. a range of tasks and optimise the system's settings based on the group tree so that the postures that best reflect the system's behaviours are selected.

**Chapter 6- Bibliography**

6.1 Books

6.2 URLs

1. Activity recognition-Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Activity\_recognition.
2. “Human Activity Recognition,” [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2015.00028/full.
3. “Research-Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/ Research.
4. “Software Requirements,” [Online]. Available: <https://www.tutorialspoint.com/software_engineering/software_requirements.htm>.
5. “Flowchart-Wikipedia,” [Online]. Available: https://en.wikipedia.org/ wik i/Flowchart.
6. “Data-Flow Diagrams-Wikipedia,” [Online]. Available: https://en.wikipedia.org/ wiki/Data-flow diagram.
7. “Residual Neural Network-Wikipedia,” [Online]. Available: <https://en.wikipedia.org/wiki/Residual_neural_network>.
8. “Overview of Resets,” [Online]. Available: https://towardsdatascience.com /an-overview-of-resnet-and-its-variants-5281e2f56035.
9. “CNN-Wikipedia,” [Online]. Available: <https://en.wikipedia.org/wiki/Convolutional_neural_network>.
10. “Python Features,” [Online]. Available: <https://www.javatpoint.com/python-features>.
11. “Advantages and Disadvantages of Python,” [Online]. Available: <https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121>.
12. “Software Testing Overview,” [Online]. Available: https://www.tutorialspoint.com/software \_engineering/software\_testing\_overview.htm.
13. “User Interface Modelling,” [Online]. Available: <https://en.wikipedia.org/wiki/User_interface_modeling>.