A Major Project Report

on

Detection of Impersonators in Examination Centre

submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

K Sai Ganesh(17211A05E5)

K Sravya Reddy (17211A05C7)

R Krishna Chaitanya (17211A05E4)

G Sudheer (18215A0530)

Under the guidance of

Ms. Priyanka S M, M.Tech,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

B.V.RAJU INSTITUTE OF TECHNOLOGY

(UGC Autonomous, Accredited by NBA & NAAC)

Vishnupur, Narspur, Medak(Dist.), Telangana State, India - 502313

2017 - 2021

B. V. Raju Institute of Technology
(UGC Autonomous, Accredited By NBA & NAAC)
Vishnupur, Narspur, Medak (Dist.),
Telangana State, India – 502313

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project entitled "**Detection of Impersonators in Examination Centre"**, being submitted by

K Sai Ganesh(17211A05E5)
K Sravya Reddy (17211A05C7)
R Krishna Chaitanya (17211A05E4)
G Sudheer (18215A0530)

In partial fulfillment of the requirements for the award of degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING to B.V.RAJU INSTITUTE OF TECHNOLOGY is a record of bonafide work carried out during a period from May 2017 to July 2021 by them under the guidance of **Ms. Priyanka S M**, Assistant Professor, CSE Department.

This is to certify that the above statement made by the students is/are correct to the best of my knowledge.

Ms. Priyanka S M
Assistant Professor

The Project Viva-Voce Examination of this team has been held on _____.

EXTERNAL EXAMINER

Dr. Ch. Madhu Babu
Professor & HoD-CSE

B. V. Raju Institute of Technology
(UGC Autonomous, Accredited By NBA & NAAC)
Vishnupur, Narspur, Medak (Dist.),
Telangana State, India – 502313

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled **"Detection of Impersonators in Examination Centre"** in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology and submitted in the Department of Computer Science and Engineering, B. V. Raju Institute of Technology, Narsapur is an authentic record of my own work carried out during a period from May 2020 to July 2021 under the guidance of **Ms. Priyanka S M**, Assistant Professor. The work presented in this project report has not been submitted by us for the award of any other degree of this or any other Institute/University.

K Sai Ganesh(17211A05E5)
K Sravya Reddy (17211A05C7)
R Krishna Chaitanya (17211A05E4)
G Sudheer (18215A0530)

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely fortunate to have got this all along the completion. Whatever we have done is due to such guidance and assistance. We would not forget to thank them.

We thank **Ms. Priyanka S M** for guiding us and providing all the support in completing this project. We are thankful to **Mr. V Pradeep Kumar**, our section project coordinator for supporting us in doing this project. We are thankful to **Mr. Karthik Kovuri**, project coordinator for helping us in completing the project in time. We thank the person who has our utmost gratitude **Dr. Ch. Madhu Babu**, Head of CSE Department.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all the staff members of CSE Department.

K Sai Ganesh(17211A05E5)
K Sravya Reddy (17211A05C7)
R Krishna Chaitanya (17211A05E4)
G Sudheer (18215A0530)

CONTENTS

DETECTING IMPERSONATORS IN EXAMINATION HALLS

ABSTRACT

Detecting impersonators in examination halls is important to provide a better way of examination handling system which can help in reducing malpractices happening in examination centers. According to the latest news reports, 56 JEE candidates who are potential impersonators were detected by a national testing agency. In order to solve this problem, an effective method is required with less manpower. With the advancement of machine learning and AI technology, it is easy to solve this problem. In this project we are developing an AI system where images of students are collected with names and hall ticket numbers are pre-trained using the KDTree algorithm and the model is saved. Whenever a student enters into classroom, the student should look at the camera and enter class, after the given time or class is filled with student's information will store in a video file with student name and hall ticket no. The video will have a user with hall ticket no and name on each face. If admin finds any unknown user tag on face admin can recheck and trace impersonators.

Existing System:

Information given in the hall ticket is used as verification to check if student is exact person or not. Manual security checks are performed with are not perfect and some time students can change image from hall ticket.

Proposed system:

In proposed system initially images of each student are collected and each dataset consists of 20 images of each student. These images are trained using KDTree algorithm using image processing technique and model is saved in system this model can be used for automatic prediction of student in exam halls from live video or images.

TEAM MEMBERS:

K Sai Ganesh(17211A05E5)

K Sravya Reddy (17211A05C7)

R Krishna Chaitanya (17211A05E4)

L Sudheer (18215A0530)

GUIDE:

Ms. Priyanka S M

CHAPTER 1

INTRODUCTION

A threat is any factor that poses any form of harm to the proper functioning and satisfying the objective of the system in a secure way ensuring confidentiality, authenticity and other necessary features. An Examination is a critical asset in the fast growing learning environment. In order to assess the threats we should understand the nature of the system and must analyze the environment where the system is deployed. Collusion is identified as the highest rated threat in an examination. Intent of most threats backtracks to the motive of cheating by the candidates for obtaining assistance during examination to enhance their chances in the examination amongst the competitors. Collusion is when the candidate invites any third party for impersonation or for aid/help the candidate in the examination. Currently authentication of the user before taking up the examination is carried out using biometric authentication and manual verification of credentials by the invigilator. This process ensures authentication at entry level. As we already discussed lack of integrity of the invigilator may increase the chance of collusion in an examination. The only way to address this issue is to conduct a runtime dynamic authentication during the examination. This paper proposes an approach to overcome this disadvantage. Facial recognition with PCA is a simple approach that can embedded to the existing system to conduct dynamic authentication of the candidate. Candidate registration photograph is used for comparison with the picture recorded while the candidate enters the examination hall. Then this recently recorded image is compared with the images captured during random intervals for identification of violations. Violation here refers to mismatch of features above a certain degree which drives to the decision that the person currently taking up the examination is not the corresponding candidate. This entire process is secured and conducted in those corresponding systems to avoid any intrusion and modification of crucial data. Violations identified by the algorithm is logged in a centralized repository under control of the examination authority. Manual verification by the invigilator is triggered to confirm the violation.

CHAPTER 2

LITERATURE SURVEY

Face detection is a computer technology that determines the location and size of human face in arbitrary (digital) image. The facial features are detected and any other objects like trees, buildings and bodies etc are ignored from the digital image. It can be regarded as a specific case of object-class detection, where the task is finding the location and sizes of all objects in an image that belong to a given class. Face detection, can be regarded as a more general case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). Basically there are two types of approaches to detect facial part in the given image i.e. feature base and image base approach. Feature base approach tries to extract features of the image and match it against the knowledge of the face features. While image base approach tries to get best match between training and testing images. FEATURE BASE APPROCH: Active Shape Model focuses on complex non-rigid features like actual physical and higher level appearance of features Means that Active Shape Models (ASMs) are aimed at automatically locating landmark points that define the shape of any statistically modelled object in an image, like the facial features such as the eyes, lips, nose, mouth and eyebrows. The training stage of an ASM involves the building of a statistical 1) facial model from a training set containing images with manually annotated landmarks. ASMs is classified into three groups i.e. snakes, PDM, Deformable templates 1.1) Snakes: The first type uses a generic active contour called snakes, first introduced by Kass et al. in 1987 Snakes are used to identify head boundaries . In order to achieve the task, a snake is first initialized at the proximity around a head boundary. It then locks onto nearby edges and subsequently assume the shape of the head. The evolution of a snake is achieved by minimizing an energy function, Esnake (analogy with physical systems), denoted asEsnake = Einternal + EExternal WhereEinternal and EExternal are internal and external energy functions.Internal energy is the part that depends on the intrinsic properties of the snake and defines its natural evolution. The typical natural evolution in snakes is shrinking or expanding. The external energy counteracts the internal energy and enables the contours to deviate from the natural evolution and eventually assume the shape of nearby features—the head boundary at a state of equilibria.Two main consideration for

forming snakes i.e. selection of energy terms and energy minimization. Elastic energy is used commonly as internal energy. Internal energy is vary with the distance between control points on the snake, through which we get contour an elastic-band characteristic that causes it to shrink or expand. On other side external energy relay on image features. Energy minimization process is done by optimization techniques such as the steepest gradient descent. Which needs highest computations. Huang and Chen and Lam and Yan both employ fast iteration methods by greedy algorithms. Snakes have some demerits like contour often becomes trapped onto false image features and another one is that snakes are not suitable in extracting non convex features. 1.2 Deformable Templates: Deformable templates were then introduced by Yuille et al. to take into account the a priori of facial features and to better the performance of snakes. Locating a facial feature boundary is not an easy task because the local evidence of facial edges is difficult to organize into a sensible global entity using generic contours. The low brightness contrast around some of these features also makes the edge detection process.Yuille et al. took the concept of snakes a step further by incorporating global information of the eye to improve the reliability of the extraction process. Deformable templates approaches are developed to solve this problem. Deformation is based on local valley, edge, peak, and brightness .Other than face boundary, salient feature (eyes, nose, mouth and eyebrows) extraction is a great challenge of face recognition.E = Ev + Ee + Ep + Ei + Einternal ; where Ev , Ee , Ep , Ei , Einternal are external energy due to valley, edges, peak and image brightness and internal energy 1.3 PDM(Point Distribution Model): Independently of computerized image analysis, and before ASMs were developed, researcher sdeveloped statistical models of shape. The idea is that once you represent shapes as vectors, you can apply standard statistical methods to them just like any other multivariateobject. These models learn allowable constellations of shape points from training examplesand use principal components to build what is called a Point Distribution Model. These have been used in diverse ways, for example for categorizing Iron Age broaches. Ideal Point Distribution Models can only deform in ways that are characteristic of the object. Cootes and his colleagues were seeking models which do exactly that so if a beard, say, covers the chin, the shape model can “override the image" to approximate the position of the chin under the beard. It was therefore natural (but perhaps only in retrospect) to adopt Point Distribution Models. This synthesis of ideas from image processing and statistical shape modelling led to the Active Shape Model.The first parametric statistical shape model for image analysis based on principal components of inter-landmark distances was presented by Cootes and Taylor. On this approach, Cootes, Taylor, and their colleagues, then released a series of papers that cumulated in what we call the classical Active Shape Model.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

IMPLEMENTATION ON (PYTHON):

What Is A Script?

Up to this point, I have concentrated on the interactive programming capability of Python. This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode.

Scripts are reusable:

Basically, a script is a text file containing the statements that comprise a Python program. Once you have created the script, you can execute it over and over without having to retype it each time.

Scripts are editable:

Perhaps, more importantly, you can make different versions of the script by modifying the statements from one file to the next using a text editor. Then you can execute each of the individual versions. In this way, it is easy to create different programs with a minimum amount of typing.

You will need a text editor:

Just about any text editor will suffice for creating Python script files. You can use Microsoft Notepad, Microsoft WordPad, Microsoft Word, or just about any word processor if you want to

Difference between a script and a program

Script:

Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, where as the applications they control are traditionally compiled to native machine code.

Program:

The program has an executable form that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executable programs are derived.

What is Python ?

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Some of the libraries used in Python:

1. Requests - The most famous http library written by kennethreitz. It's a must have for every python developer.

2. Scrapy - If you are involved in webscraping then this is a must have library for you. After using this library you won't use any other.

3. wxPython - A gui toolkit for python. I have primarily used it in place of tkinter. You will really love it.

4. Pillow - A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.

5. SQLAlchemy - A database library. Many love it and many hate it. The choice is yours.

6. BeautifulSoup - I know it's slow but this xml and html parsing library is very useful for beginners.

7. Twisted - The most important tool for any network application developer. It has a very beautiful api and is used by a lot of famous python developers.

8. NumPy - How can we leave this very important library ? It provides some advance math functionalities to python.

9. SciPy - When we talk about NumPy then we have to talk about scipy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.

10. matplotlib - A numerical plotting library. It is very useful for any data scientist or any data analyzer.

11. Pygame - Which developer does not like to play games and develop them ? This library will help you achieve your goal of 2d game development.

12. Pyglet - A 3d animation and game creation engine. This is the engine in which the famous python port of minecraft was made

13. pyQT - A GUI toolkit for python. It is my second choice after wxpython for developing GUI's for my python scripts.

14. pyGtk - Another python GUI library. It is the same library in which the famous Bittorrent client is created.

15. Scapy - A packet sniffer and analyzer for python made in python.

16. pywin32 - A python library which provides some useful methods and classes for interacting with windows.

17. nltk - Natural Language Toolkit – I realize most people won't be using this one, but it's generic enough. It is a very useful library if you want to manipulate strings. But it's capacity is beyond that.

18. nose - A testing framework for python. It is used by millions of python developers. It is a must have if you do test driven development.

19. SymPy - SymPy can do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.

20. IPython - I just can't stress enough how useful this tool is. It is a python prompt on steroids. It has completion, history, shell capabilities, and a lot more.

CHAPTER 4

DESIGN DOCUMENT

4.1 UML DIAGRAMS

4.1.1 CLASS DIAGRAM -

Class diagram 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.

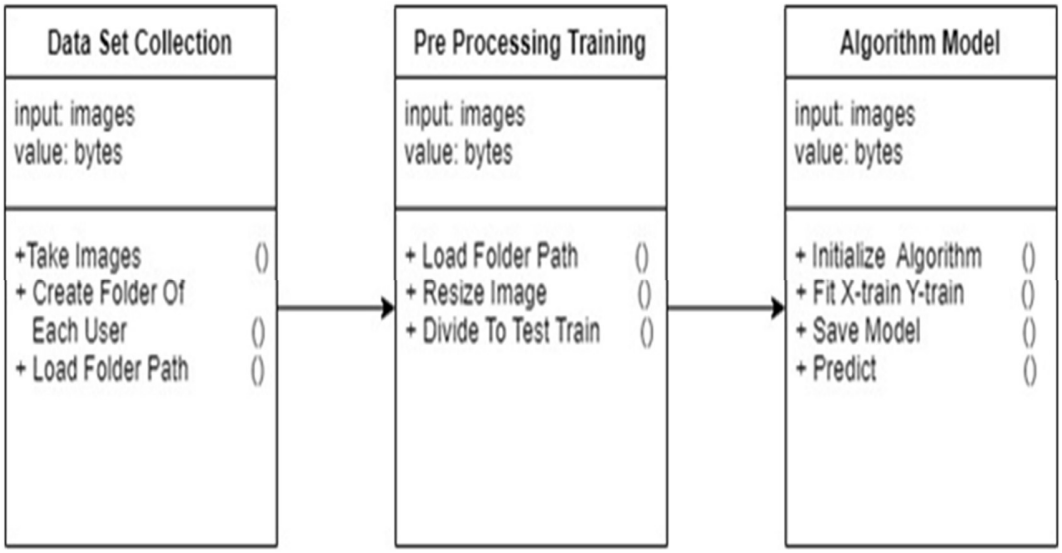


FIG 4.1.1 :- CLASS DIAGRAM

4.1.2 USECASE DIAGRAM -

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases.So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

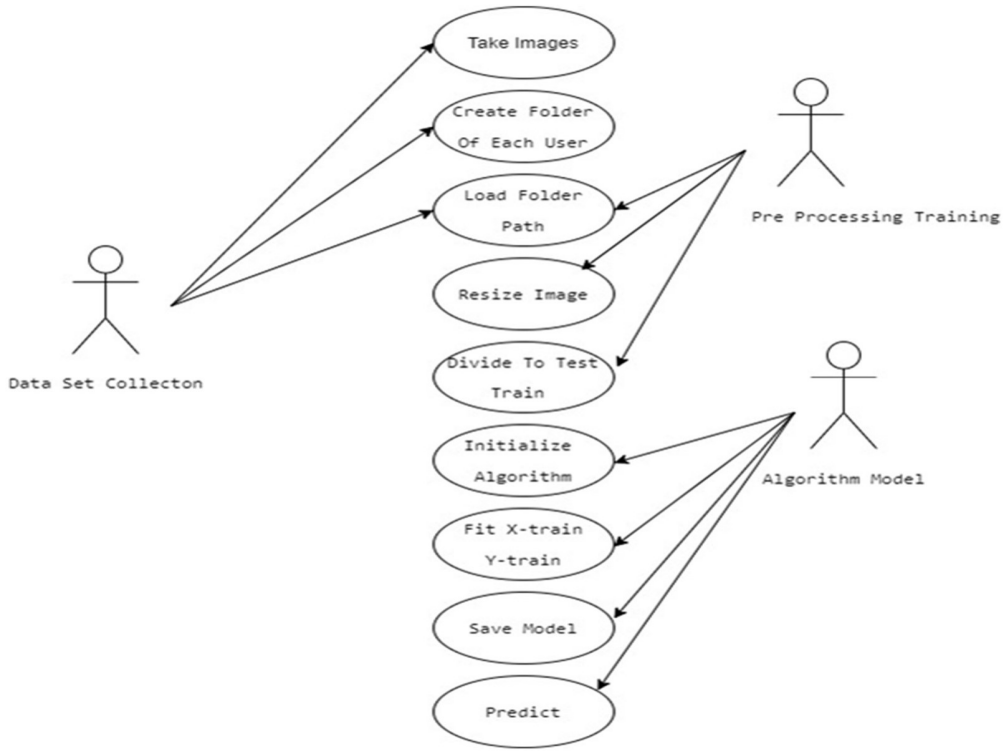


FIG 4.1.2 :- USECASE DIAGRAM

4.1.3 SEQUENCE DIAGRAM -

A **sequence diagram** in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

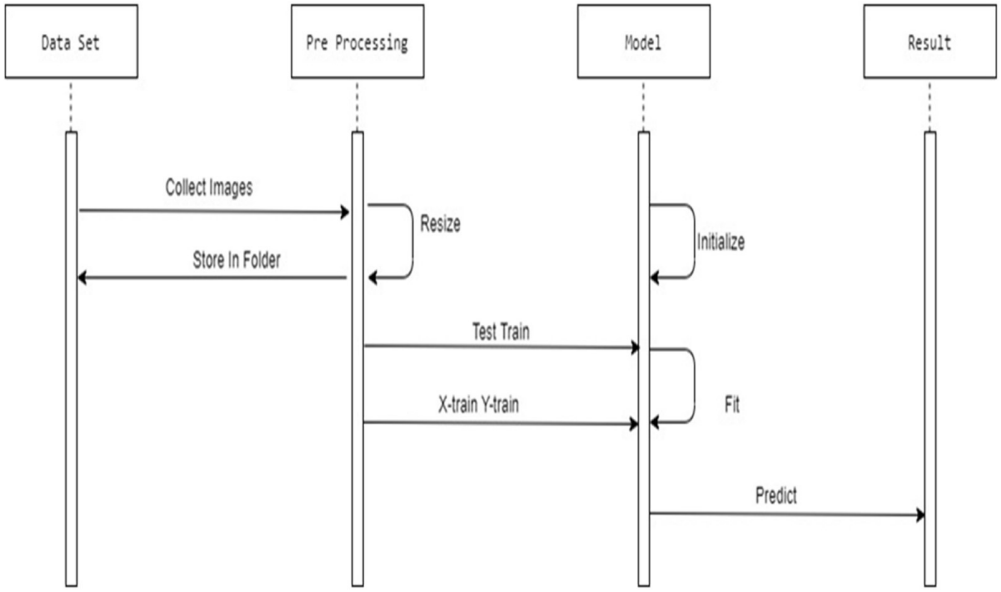


FIG 4.1.3 :- SEQUENCE DIAGRAM

4.1.4 ACTIVITY DIAGRAM -

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

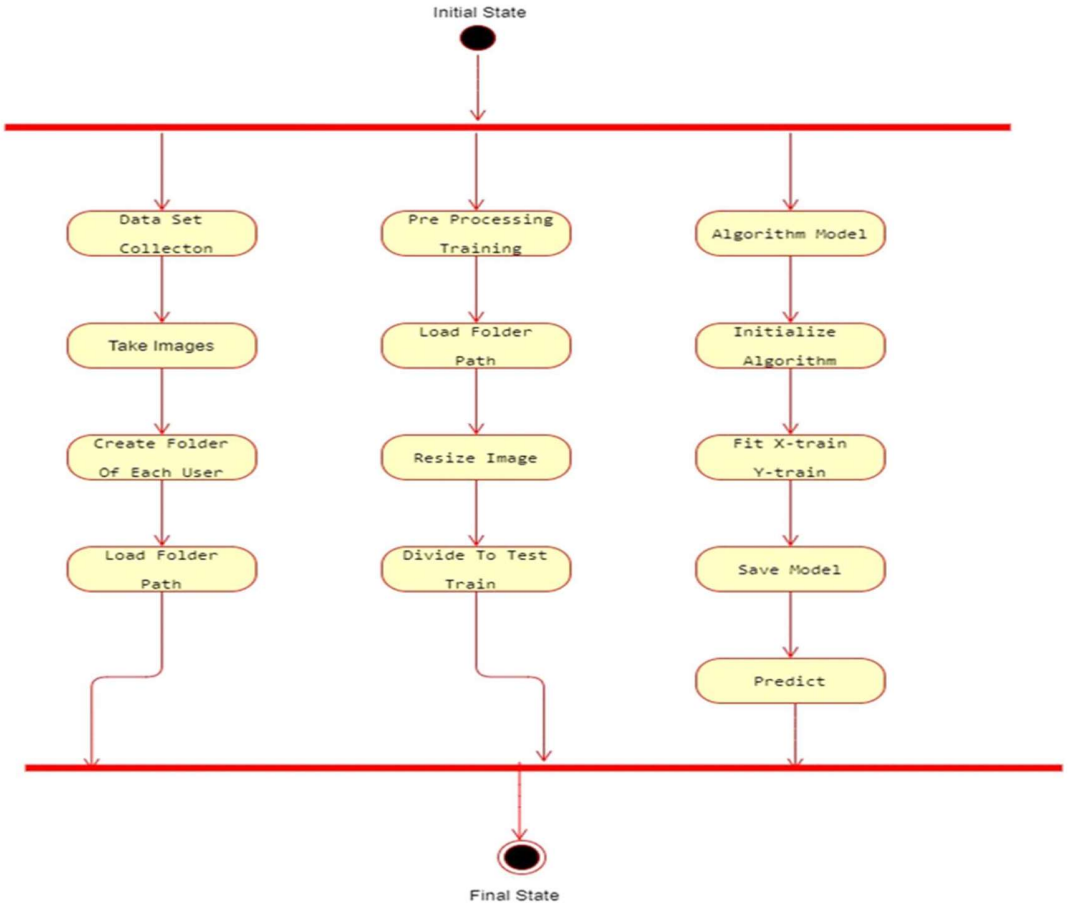


FIG 4.1.4 :- ACTIVITY DIAGRAM

4.2 ARCHITECTURE MODEL

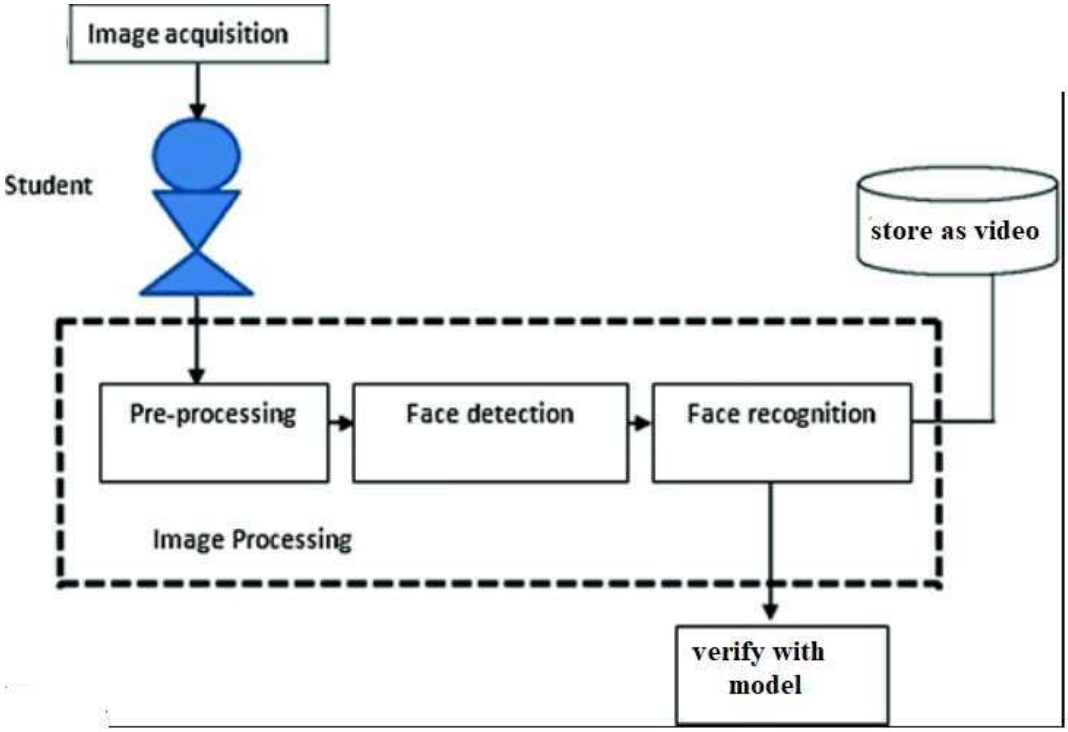


FIG 4.2 :- ARCHITECTURE MODEL

Image Acquisition:

Digital image Processing. In image processing, it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing. It is the first step in the workflow sequence because, without an image, no processing is possible.

Pre-processing:

Pre-processing is a common name for operations with images at the lowest level of abstraction — both input and output are intensity images. These iconic images are of the same kind as the original data captured by the sensor, with an intensity image usually represented by a matrix of image function values (brightnesses). The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g. rotation, scaling, translation) are classified among pre-processing methods here since similar techniques are used.

Face Detection:

The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. There are two kinds of methods that are currently popular in developed face recognition pattern namely, Eigenface method and Fisherface method. Facial image recognition Eigenface method is based on the reduction of facedimensional space using Principal Component Analysis (PCA) for facial features. The main purpose of the use of PCA on face recognition using Eigen faces was formed (face space) by finding the eigenvector corresponding to the largest eigenvalue of the face image. The area of this project face detection system with face recognition is Image processing. The software requirements for this project is matlab software. Keywords: face detection, Eigen face, PCA, matlab Extension: There are vast number of applications from this face detection

project, this project can be extended that the various parts in the face can be detect which are in various directions and shapes.

FACE RECOGNITION:

There are two predominant approaches to the face recognition problem: Geometric (feature based) and photometric (view based). As researcher interest in face recognition continued, many different algorithms were developed, three of which have been well studied in face recognition literature. Recognition algorithms can be divided into two main approaches: 1. Geometric: Is based on geometrical relationship between facial landmarks, or in other words the spatial configuration of facial features. That means that the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are classified on the basis of various geometrical distances and angles between features. (Figure 3) 2. Photometric stereo: Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normals (Zhao and Chellappa, 2006) (Figure 2) Popular recognition algorithms include: 1. Principal Component Analysis using Eigenfaces, (PCA) 2. Linear Discriminate Analysis, 3. Elastic Bunch Graph Matching using the Fisherface algorithm.

CHAPTER 5

CODE

5.1 IMAGE PROCESSING -

```
def _encode_faces(imagePath, detection_method):
        known_encodings = []
        known_names = []
        try:
            name = imagePath.split(os.path.sep)[-2]
            image = cv2.imread(imagePath)
            rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
            boxes = face_recognition.face_locations(rgb,model=detection_method)
            encodings = face_recognition.face_encodings(rgb, boxes
          for encoding in encodings:
            known_encodings.append(encoding)
            known_names.append(name)
            return known_encodings, known_names

    except Exception as ex:
            print("exception while encoding image %s : %s" % (imagePath, ex))
            return [], []
```

5.2 K-D TREE -

```
if args["fast_nn"]:
        encoding_structure = constants.ENC_KDTREE
        known_encodings=KDTree(np.asarray(known_encodings),leaf_size=constants.LE
AF_SIZE_KDTREE)
data = {constants.KNOWN_ENCODINGS: known_encodings,
        constants.KNOWN_NAMES: known_names,
        constants.ENCODING_STRUCTURE:  encoding_structure}
```

5.3 TRAINING THE DATA -

```
print("[INFO] quantifying faces...")
image_paths = list(paths.list_images(args["dataset"]))
known_encodings = []
known_names = []
  mp_batch_size = args["cores"] * 10
encode_images_with_detection_method = functools.partial(_encode_faces,
                                                detection_method=args["detection_method"])
```

5.4 LOADING FACES AND INITIALIZING VIDEO STREAM -

```
print("[INFO] loading encodings...")
data = pickle.loads(open(args["encodings"], "rb").read())
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
```

```
writer = None

time.sleep(2.0)


faceRec = FaceRec()
```

5.5 DETAILS OF BOX -

```
for ((top, right, bottom, left), name) in zip(boxes, names):
                top = int(top * r)
                right = int(right * r)
                bottom = int(bottom * r)
                left = int(left * r)


  cv2.rectangle(frame, (left, top), (right, bottom),
                        (0, 255, 0), 2)
                y = top - 15 if top - 15 > 15 else top + 15
                cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
                        0.75, (0, 255, 0), 2)
```

CHAPTER 6

IMPLEMENTATION

DATA SET COLLECTION:

Students data set can be collected manually whicle issusing halltickets and store in each folder. Minimum 30 to 50 images need to be collected rom each student.

ENCODING DATASET:

In this stage face recognition, pickle, opencv, sklearn libraries are initialized and each image is looped from the folder using batching and multiprocessing.

- load the input image and convert it from BGR
- detect the (x, y)-coordinates of the bounding boxes
- corresponding to each face in the input image compute the facial embedding for the face

K-D TREE ALGORITHM:

Encoded images are given to kdd tree algorithm and model is saved in pickle format.

K-d tree data structure has been used as a data structure for overcoming increased processing times caused by the addition of features into the database. An alternative approach is establishing a balanced k-d tree

PREDICTION:

In this stage input values are taken from image or video and compared with model and results are predicted and output is stored in image or video.

CHAPTER 7

TESTING

TESTING CODE -

- Code is usually developed in a file using an editor.
- To test the code, import it into a Python session and try to run it.
- Usually there is an error, so you go back to the file, make a correction, and test again.
- This process is repeated until you are satisfied that the code works.
- The entire process is known as the development cycle.
- There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid.
- This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error.

WEB FRAMEWORK LEARNING CHECKLIST -

1. Choose a major Python web framework and stick with it. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.
2. Work through a detailed tutorial found within the resources links on the framework's page.
3. Study open source examples built with your framework of choice so you can take parts of those projects and reuse the code in your application.
4. Build the first simple iteration of your web application then go to the deployment section to make it accessible on the web.

CHAPTER 8

OUTPUT SCREEN

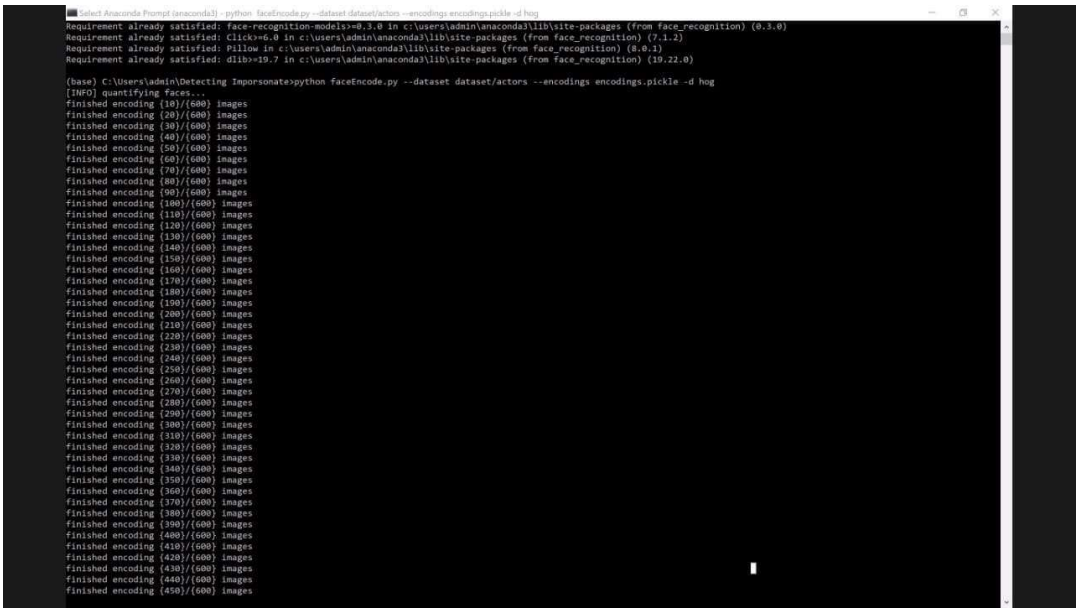


FIG 8.1 : QUANTIFYING FACES

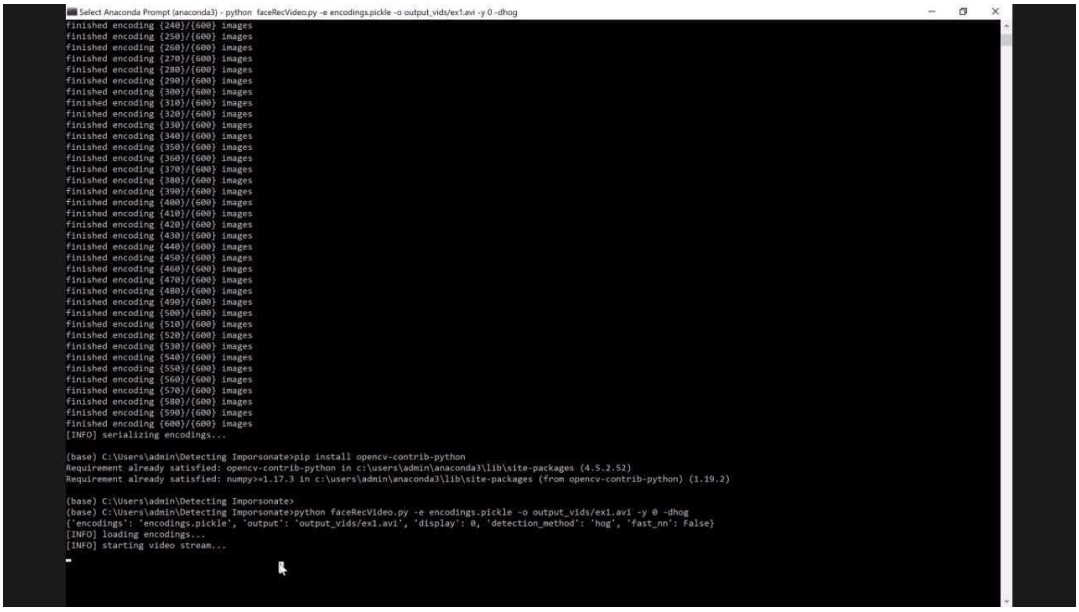
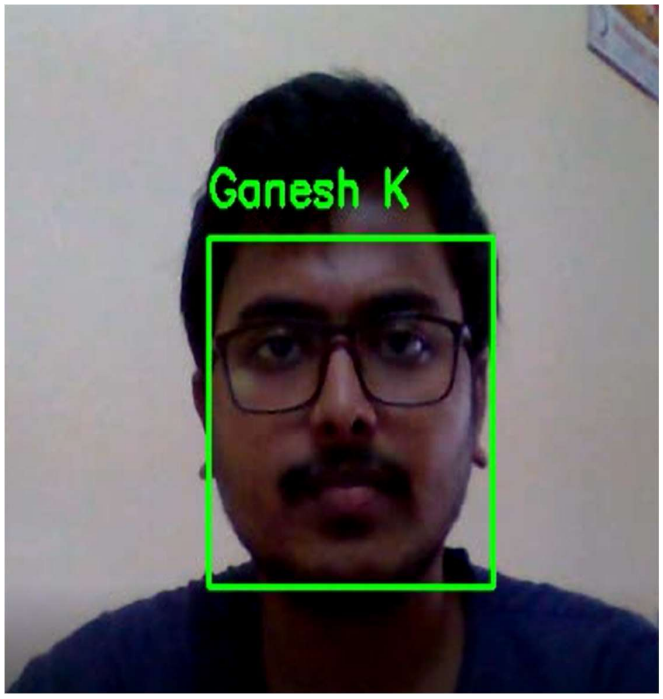


FIG 8.2: VIDEO STREAM



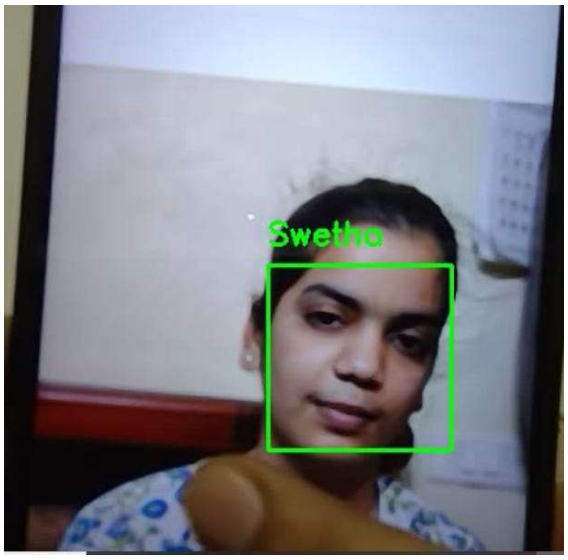


FIG 8.3: SCREENSHOT OF OUTPUT VIDEO

CHAPTER 9

FUTURE ENHANCEMENT

IMPLECATIONS FOR FURTHER STUDY -

- To make this more efficient, we will need more data.

- In future we will build a complete open source working platform with huge amounts of image data.

- In future, we would be able to detect all the students present in the classroom at once.

CHAPTER 10

CONCLUSION

The computational models, which were implemented in this project, were chosen after extensive research, and the successful testing results confirm that the choices made by the researcher were reliable.The system with manual face detection and automatic face recognition did not have a recognition accuracy over 90%, due to the limited number of eigenfaces that were used for the PCA transform. This system was tested under very robust conditions in this experimental study and it is envisaged that real-world performance will be far more accurate.The fully automated frontal view face detection system displayed virtually perfect accuracy and in the researcher's opinion further work need not be conducted in this area. The fully automated face detection and recognition system was not robust enough to achieve a high recognition accuracy. The only reason for this was the face recognition subsystem did not display even a slight degree of invariance to scale, rotation or shift errors of the segmented face image. However, if some sort of further processing, such as an eye detection technique, was implemented to further normalise the segmented face image, performance will increase to levels comparable to the manual face detection and recognition system. Implementing an eye detection technique would be a minor extension to the implemented system and would not require a great deal of additional research.All other implemented systems displayed commendable results and reflect well on the deformable template and Principal Component Analysis strategies.The most suitable real-world applications for face detection and recognition systems are for mugshot matching and surveillance. There are better techniques such as iris or retina recognition and face recognition using the thermal spectrum for user access and user verification applications since these need a very high degree of accuracy.The real-time automated pose invariant face detection and recognition system proposed in chapter seven would be ideal for crowd surveillance applications. If such a system were widely implemented its potential for locating and tracking suspects for law enforcement agencies is immense. The implemented fully automated face detection and recognition system (with an eye detection system) could be used for simple surveillance applications such as ATM user security. If a face recognition system can reduce

the number of images that a human operator has to search through for a match from 10000 to even a 100, it would be of incredible practical use in law enforcement. The automated vision systems implemented in this thesis did not even approach the performance, nor were they as robust as a human's innate face recognition system. However, they give an insight into what the future may hold in computer vision. So everyone should need to work with image classifier.It may change the entire concept what you have been understood earlier.

CHAPTER 11

REFERENCES

- T.-H. Kim, D.-C. Park, D.-M. Woo, T. Jeong, and S.-Y. Min, "Multi-class classifier-based adaboost algorithm," in Proceedings of the Second Sinoforeign-interchange Conference on Intelligent Science and Intelligent Data Engineering, ser. IScIDE'11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 122– 127.

- P. Viola and M. J. Jones, "Robust real-time face detection," Int. J. Comput. Vision, vol. 57, no. 2, pp. 137–154, May 2004.

- P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, Dec 2001, pp. I–I.

- J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, and J. Feng, "Towards real world human parsing: Multiple-human parsing in the wild," CoRR, vol. abs/1705.07206.

- J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, and J. Feng, "Towards real world human parsing: Multiple-human parsing in the wild," CoRR, vol. abs/1705.07206.