**A MINI PROJECT REPORT**

**On**

**Face Detection and Recognition in organic video: A comparative study for sport celebrities database – Using Deep Learning**

**Submitted to Jawaharlal Nehru Technological university for the partial**

**Fulfillment of the Requirement for the Award of the Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

|  |  |  |
| --- | --- | --- |
| **N BHASKAR NAYAK** | **-** | **20Q91A05I9** |
| **B VINEELA** | **-** | **20Q91A05J2** |
| **M SUNAYANA** | **-** | **20Q91A05I4** |
| **K NIKHIL VERMA** | **-** | **20Q91A05L2** |

Under the guidance of

**Mrs. K ASHWINI**

**ASSISTANT PROFESSOR**

****

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**MALLA REDDY COLLEGE OF ENGINEERING**

**(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)**

Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New

Delhi ISO 9001:2015 certified Institution

Maisammaguda, Dhulapally (Post via Kompally ), Secunderabad – 500100

**2023 – 2024**

**MALLA REDDY COLLEGE OF ENGINEERING**

**(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)**

Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New

Delhi ISO 9001:2015 certified Institution

Maisammaguda, Dhulapally (Post via Kompally ), Secunderabad – 500100

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

This is to certify that the Minor Project report on “Face detection and recognition in organic video: A comparative study for sport celebrities database – using Deep Learning” Is successfully done by the following students of Department of Computer Science and Engineering of our college in partial fulfilment of the requirement for the award of B.Tech degree in the year 2023-2024. The results embodied in this report have not been submitted to any other university for the award of any diploma or degree.

|  |  |
| --- | --- |
| N BHASKAR NAYAK : | 20Q91A05I9 |
| B VINEELA : | 20Q91A05J2 |
| M SUNAYANA : | 20Q91A05I4 |
| K NIKHIL VERMA : | 20Q91A05L2 |

Submitted for the viva voce examination held on:

|  |  |  |
| --- | --- | --- |
| **INTERNAL GUIDE**  **Mrs. K ASHWINI**  **Asst Professor** | **H O D**  **Dr. G Radha Devi**  **Asst Professor** | **PRINCIPAL**  **Dr. Ashok Reddy**  **Professor** |

|  |  |
| --- | --- |
| **Internal Examiner** | **External Examiner** |

**DECLARATION**

We, the final year students are hereby declaring the the minor project report entitled "Face detection and recognition in organic video: A comparative study for sport celebrities database – using Deep Learning " has done by us under the guidance of Mrs. K Ashwini Assistant Professor, Department of CSE is submitted in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE OF ENGINEERING.**

The Results embedded in this project report have not been submitted to any other University or institute for the award of any degree or diploma.

Signature of the Candidate

|  |  |
| --- | --- |
| **N BHASKAR NAYAK** | **20Q91A05I9** |
| **B VINEELA** | **20Q91A05J2** |
| **M SUNAYANA** | **20Q91A05I4** |
| **K NIKHIL VERMA** | **20Q91A05l2** |

**DATE:**

**PLACE: Maisammaguda**

# ACKNOWLEDGEMENT

First and foremost, we would like to express our immense gratitude towards our institution Malla Reddy College of Engineering, which helped us to attain profound technical skills in the field of Computer Science & Engineering, there by fulfilling our most cherished goal.

We are pleased to thank **Sri Ch. Malla Reddy**, our Founder, Chairman **MRGI**, **Sri Ch. Mahender** **Reddy**, Secretary, **MRGI** for providing this opportunity and support throughout the course.

It gives us immense pleasure to acknowledge the perennial inspiration of **Dr. Ashok Reddy** our beloved principal for his kind co-operation and encouragement in bringing out this task.

We would like to thank **Dr. T. V. Reddy** our vice principal, **Dr. G. Radha Devi** HOD, CSE Department for their inspiration adroit guidance and constructive criticism for successful completion of our degree.

We would like to thank **Mrs. K Ashwini** Assistant Professor our internal guide, for his valuable suggestions and the guidance during the exhibition and completion of the project.

Finally, we avail this opportunity to express our deep gratitude to all staff who have contribute their valuable assistance and support making our project success.

**N. BHASKAR NAYAK - 20Q91A05I9**

**B. VINEELA - 20Q91A05J2**

**M. SUNAYANA - 20Q91A05I4**

**K. NIKHIL VERMA - 20Q91A05L2**

**INDEX**

**TITLES PAGE NO**

**CONTENTS**

[ABSTRACT vi](#_TOC_250016)

1. [INTRODUCTION 1](#_TOC_250015)
2. LITERATURE SURVEY 5
   1. 2.1 KEY-DEDUPLICATION WITH IBBE 5
   2. 2.2 SERVER LESS DISTRIBUTEDFILESYSTEM 6
   3. 2.3 [THE GOOGLEFILESYSTEM 7](#_TOC_250010)
   4. 2.4 CONVERGENTKEYMANAGEMENT 8
   5. 2.5 [SOFTWAREENVIRONMENT 9](#_TOC_250009)
   6. 2.6 [WHY CHOOSEPYTHON 10](#_TOC_250008)
3. [SYSTEM](#_TOC_250007) ANALYSIS 14
   1. 3.1 [EXISTING SYSTEM 16](#_TOC_250006)
   2. 3.2 [PROPOSEDSYSTEM 16](#_TOC_250005)
4. [FEASIBILITYSTUDY 17](#_TOC_250004)
   1. 4.1 [ECONOMICALFEASIBILITY 17](#_TOC_250003)
   2. 4.2 [TECHNICALFEASIBILITY 17](#_TOC_250002)
   3. 4.3 [SOCIALFEASIBILITY 18](#_TOC_250001)
5. [SYSTEMREQUIREMENTS 19](#_TOC_250000)

## SYSTEMDESIGN 20

* 1. 6.1 SYSTEMARCHITECTURE 20
  2. 6.2 DATAFLOW DIAGRAM 20
  3. 6.3 UMLDIAGRAMS 22

## IMPLEMENTATION 27

* 1. 7.1 MODULES 27
  2. 7.2 SAMPLE CODE 28

## SYSTEMTESTING 29

* 1. 8.1 UNITTESTING 31
  2. 8.2 INTEGRATION TESTING 32
  3. 8.3 ACCEPTANCETESTING 33

## INPUT DESIGN ANDOUTPUTDESIGN 34

* 1. 9.1 INPUTDESIGN 34
  2. 9.2 OUTPUTDESIGN 35

## SCREENSHOTS 37

## FUTURE SCOPE 50

## CONCLUSION 51

## BIBLOGRAPHY 52

**ABSTRACT**:

Now-a-days with the continued development of artificial intelligence facial emotion recognition has become more popular. The emotion recognition plays a major role in interaction technology. In interaction technology the verbal components only play a one third of communication and the non-verbal components plays a two third of communication. A facial emotion recognition (FER) method is used for detecting facial expressions. Facial expression plays a major role in expressing what a person feels and it expresses inner feeling and his or her mental situation or human perspective. This paper aims to identify basic human emotions with the combination of gender classification and age estimation. The facial emotions such as happy, sad, angry, fear, surprised, neutral emotions are considered as basic emotions. Here proposes a real time facial emotion recognition system based on You Look Only Once (YOLO) version 2 architecture and a squeezenet architecture. The yolo architecture is a real time object detection system. Here it used for identify and detect faces in real time. These images are captured by using anchor boxes for accuracy. The second architecture is squeezenet and is used for gender classification and age estimation. It provides significant, accurate object detection and extracts high-level features that help to achieve tremendous performance to classify the image and detecting objects. Both the architectures provide accurate result than other methods with the large no of hidden layers and cross validation in the neural network.

**CHAPTER - 1:**

**INTRODUCTION**

**1. INTRODUCTION**

A face detection includes classifying image into two classes: one with faces (targets), and other containing the background (clutter) which needs to be removed. Commonalities exist between faces, they vary differently in terms of age, skin color and facial expression, this becomes difficult due to this commonality. The further problem is complicated by differing lighting conditions, image qualities and geometries, partial occlusion and disguise is also a possibility. A face detector should be able to detect the presence of any face under different set of lighting conditions in any background condition. The face detection analysis can be broken into two tasks. The first task is a classification task that takes some random regions of image as input and outputs a binary value of yes or no, indicating if there are any faces present in the image. The other task is the face localization task which is to take an image as input and output the location of any face or faces within that image as some bounding box/boxes with (x, y, width, height). Smart robots can be built by automatic facial expression application. These bots can be used in various applications like interactive games and service center. There are six universal expressions according to Ekman they are fear, disgust surprise, anger, sadness and happiness. Face variances can be observed to recognize these expressions. For example, we can say a person is happy which can be identified as a gesture of smile by tightened eyelids and raised lips corners. A person’s internal states, social communication and intentions are indicated by change in facial expressions. Many applications in many areas like human emotions analysis, natural Human computer interaction, image retrieval and talking bots have a large effect on them by automatic facial expression detection. Face Recognition with Histogram of Oriented Gradients using CNN detection has been an impacting issue in the technological community as human beings fined facial expressions one of the most natural and powerful means to express their intentions and emotions. Last stage of the system is facial expression detection. There are basically three steps in training procedure in expression recognition systems named as feature learning, classifier construction and feature selection. Feature learning stage is first, feature selection is second and the last one is classifier construction. Only learned facial expressions variations among all features are extracted after feature learning stage. Facial expression is then represented by the best features which are chosen by feature selection. Not only maximizing inter class variation but they also should try to minimize the intra class variations of expressions not only maximizing inters class variation but they also should minimize the intra class variations of expressions. Because same expressions of different individuals in image are far from each other in pixel’s space so minimizing the intra class variation of expressions is a problem. Techniques that can be used for facial detection are YOLO, SDD, RCNN, Faster RCNN.

**CHAPTER - 2:**

**LITERATURE SURVEY**

**2. LITERATURE SURVEY**

**2.1 Facial Expression Recognition with Histogram of Oriented Gradients using CNN.**

**Authors:** Fayyaz Ali,Sahar Zafar,Irfan Ali,Subhash Guriro,Asif KhanAdnan Zaidi

**Abstract:** A new method is introduced in this study for Facial expression recognition using FER2013 database consisting seven classes consisting (Surprise, Fear, Angry, Neutral, Sad, Disgust, Happy) in past few decades, Exploration of methods to recognize facial expressions have been active research area and many applications have been developed for feature extraction and inference. However, it is still challenging due to the high-intra class variation. Methods/Statistical Analysis: we deeply analyzed the accuracy of both handcrafted and leaned aspects such as HOG. This study proposed two models; (1) FER using Deep Convolutional Neural Network (FER-CNN) and (2) Histogram of oriented Gradients based Deep Convolutional Neural Network (FER-HOGCNN). the training and testing accuracy of FER-CNN model set 98%, 72%, similarly Losses were 0.02, 2.02 respectively. On the other side, the training and testing accuracy of FER-HOGCNN model set 97%, 70%, similarly Losses were 0.04, 2.04. Findings: It has been found that the accuracy of FER-HOGCNN model is good overall but comparatively not better than Simple FER-CNN. In dataset the quality of images are low and small dimensions, for that reason, the HOG loses some important features during training and testing. Application/Improvements: The study helps for improving the FER System in image processing and furthermore, this work shall be extended in future, and order to extract the important features from images by combining LBP and HOG operator using Deep Learning models.

**2.2 Active Clustering with Ensembles for Social structure extraction**

**Authors:** J. R. Barr, L. A. Cament, K. W. Bowyer, and P. J. Flynn

**Abstract:** We introduce a method for extracting the social network structure for the persons appearing in a set of video clips. Individuals are unknown, and are not matched against known enrollments. An identity cluster representing an individual is formed by grouping similar-appearing faces from different videos. Each identity cluster is represented by a node in the social network. Two nodes are linked if the faces from their clusters appeared together in one or more video frames. Our approach incorporates a novel active clustering technique to create more accurate identity clusters based on feedback from the user about ambiguously matched faces. The final output consists of one or more network structures that represent the social group(s), and a list of persons who potentially connect multiple social groups. Our results demonstrate the efficacy of the proposed clustering algorithm and network analysis techniques.

**2.3 Fast human detection using a cascade of histograms of oriented gradients**

**Authors:** Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan

**Abstract:** We integrate the cascade-of-rejectors approach with the Histograms of Oriented Gradients (HoG) features to achieve a fast and accurate human detection system. The features used in our system are HoGs of variable-size blocks that capture salient features of humans automatically. Using AdaBoost for feature selection, we identify the appropriate set of blocks, from a large set of possible blocks. In our system, we use the integral image representation and a rejection cascade which significantly speed up the computation. For a 320 × 280 image, the system can process 5 to 30 frames per second depending on the density in which we scan the image, while maintaining an accuracy level similar to existing methods.

**2.4 Local directional ternary pattern for facial expression recognition**

**Authors:** Ryu, A. R. Rivera, J. Kim, and O. Chae

**Abstract:** This paper presents a new face descriptor, local directional ternary pattern (LDTP), for facial expression recognition. LDTP efficiently encodes information of emotion-related features (ı.e., eyes, eyebrows, upper nose, and mouth) by using the directional information and ternary pattern in order to take advantage of the robustness of edge patterns in the edge region while overcoming weaknesses of edge-based methods in smooth regions. Our proposal, unlike existing histogram-based face description methods that divide the face into several regions and sample the codes uniformly, uses a two-level grid to construct the face descriptor while sampling expression-related information at different scales. We use a coarse grid for stable codes (highly related to non-expression), and a finer one for active codes (highly related to expression). This multi-level approach enables us to do a finer grain description of facial motions while still characterizing the coarse features of the expression. Moreover, we learn the active LDTP codes from the emotion-related facial regions. We tested our method by using person-dependent and independent cross-validation schemes to evaluate the performance. We show that our approaches improve the overall accuracy of facial expression recognition on six data sets.

**2.5 SOFTWARE ENVIRONMENT**

**Python** is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt. Sources [newsgroup](https://en.wikipedia.org/wiki/Usenet) in 1991, and version 1.0 was released in 1994.

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been back ported to Python 2. But in general, they remain not quite compatible.

Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official [End of Life date of January 1, 2020](https://pythonclock.org/) has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python, it is recommended that you focus on Python 3, as this tutorial will do.

Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe [Monty Python’s Flying Circus](https://en.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus), of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

**2.6 WHY CHOOSE PYTHON**

If you’re going to write programs, there are literally dozens of commonly used languages to choose from. Why choose Python? Here are some of the features that make Python an appealing choice.

**Python is Popular**

Python has been growing in popularity over the last few years. The 2018 [Stack Overflow Developer Survey](https://insights.stackoverflow.com/survey/2018) ranked Python as the 7th most popular and the number one most wanted technology of the year. [World-class software development countries around the globe use Python every single day.](https://realpython.com/world-class-companies-using-python/)

According to [research by Dice](https://insights.dice.com/2016/02/01/whats-hot-and-not-in-tech-skills/) Python is also one of the hottest skills to have and the most popular programming language in the world based on the [Popularity of Programming Language Index](https://pypl.github.io/PYPL.html).

Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you’d like to dig deeper into [Python salary statistics and job opportunities, you can do so here](https://dbader.org/blog/why-learn-python).

**Python is interpreted**

Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer’s processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.

One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

### Python is Free

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes.

A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

### Python is Portable

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

### Python is Simple

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way.

A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect.

Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read.

But It’s Not That Simple For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and [object-oriented programming](https://realpython.com/python3-object-oriented-programming/).

Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.

Python accomplishes what many programming languages don’t: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

## Conclusion

This section gave an overview of the **Python** programming language, including:

* A brief history of the development of Python
* Some reasons why you might select Python as your language of choice

Python is a great option, whether you are a beginning programmer looking to learn the basics, an experienced programmer designing a large application, or anywhere in between. The basics of Python are easily grasped, and yet its capabilities are vast. Proceed to the next section to learn how to acquire and install Python on your computer.

**Python** is an [open source](https://simple.wikipedia.org/wiki/Open_source) [programming language](https://simple.wikipedia.org/wiki/Programming_language) that was made to be easy-to-read and powerful. A [Dutch](https://simple.wikipedia.org/wiki/Netherlands) programmer named [Guido van Rossum](https://simple.wikipedia.org/wiki/Guido_van_Rossum) made Python in 1991. He named it after the television show [Monty Python's Flying Circus](https://simple.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus). Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be [compiled](https://simple.wikipedia.org/wiki/Compiled_language) to run. A program called an [interpreter](https://simple.wikipedia.org/wiki/Interpreter_(computing)) runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like [C](https://simple.wikipedia.org/wiki/C_(programming_language)), because it is not running [machine code](https://simple.wikipedia.org/wiki/Machine_code) directly.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

Python drew inspiration from other programming languages like C, [C++](https://simple.wikipedia.org/wiki/C%2B%2B), [Java](https://simple.wikipedia.org/wiki/Java_(programming_language)), [Perl](https://simple.wikipedia.org/wiki/Perl), and [Lisp](https://simple.wikipedia.org/wiki/LISP).

Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a [semicolon](https://simple.wikipedia.org/wiki/Semicolon) at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language.

**Python use [change / change source]**

Python is used by hundreds of thousands of programmers and is used in many

places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.

Its [standard library](https://simple.wikipedia.org/w/index.php?title=Standard_library&action=edit&redlink=1) is made up of many [functions](https://simple.wikipedia.org/wiki/Computable_function) that come with Python when it is installed. On the [Internet](https://simple.wikipedia.org/wiki/Internet) there are many other [libraries](https://simple.wikipedia.org/w/index.php?title=Library_(computing)&action=edit&redlink=1) available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

* Web development
* Scientific programming
* Desktop [GUIs](https://simple.wikipedia.org/wiki/GUI)
* Network programming
* [Game](https://simple.wikipedia.org/wiki/Video_game) programming

**Algorithm:**

**Object Recognition** is a technology that lies under the broader domain of Computer Vision. This technology is capable of identifying objects that exist in images and videos and tracking them. Object Recognition also known as **Object Detection**, has various applications like face recognition, vehicle recognition, pedestrian counting, self-driving vehicles, security systems, and a lot more.

The two significant objectives of object recognition involve:

1. **Identification** of all objects that exist in an image
2. **Filtration** of the object that seeks attention

In the following tutorial, we will understand how to perform Object Recognition in the Python programming language using the **ImageAI** library. We will create a basic object recognition model using the **ImageAI** library in Python by the end of this tutorial.

So, let's get begun.

## Deep Learning for Object Recognition

Techniques of Deep learning have been shown state of the art for different problems related to Object Recognition. Some of the generally used approaches of deep learning for object recognition are as follows:

1. **ImageAI**
2. **Single Shot Detectors**
3. **YOLO (You Only Look Once)**
4. **Region-based Convolutional Neural Networks**

However, in this tutorial, we will understand what **ImageAI** is and how we can use it in performing Object Recognition.

## Understanding the ImageAI library

Python offers a library built to empower programmers and developers for building applications and systems with self-contained deep learning and Computer Vision capabilities with the help of some lines of simple coding script. **ImageAI** consists of Python implementation of nearly all state-of-the-art deep learning algorithms such as **RetinaNet, YOLOv3**, and **TinyYOLOv3**.

**ImageAI** makes use of several APIs that work offline - it has object detection, video detection, and object tracking APIs that can be called without accessing the Internet. **ImageAI** uses a pre-trained model and can easily be customized.

The **Object Detection** class of the **ImageAI** library consists of methods in order to perform object detection on any image or set of images with the help of pre-trained models. With **ImageAI**, we can detect and recognize eighty distinct types of common, everyday objects.

## Setting up the Environment

In this section of the tutorial, we will consider working through the installation of required libraries, including **ImageAI**.

In order to utilize **ImageAI**, we have to install some dependencies. The initial step is to have Python installed on the system. We can download and install Python 3 from Python's official website: <https://www.python.org/>.

Once we have installed Python on the system successfully, we have to install the following dependencies with the help of the **pip** installer:

**CHAPTER - 3:**

**SYSTEM ANALYSIS**

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM:**

The emotion recognition plays a major role in interaction technology. In interaction technology the verbal components only play a one third of communication and the non-verbal components plays a two third of communication. A facial emotion recognition (FER) method is used for detecting facial expressions. Facial expression plays a major role in expressing what a person feels and it expresses inner feeling and his or her mental situation or human perspective

**3.2 PROPOSED SYSTEM:**

This paper aims to identify basic human emotions with the combination of gender classification and age estimation. The facial emotions such as happy, sad, angry, fear, surprised, neutral emotions are considered as basic emotions. Here proposes a real time facial emotion recognition system based on You Look Only Once (YOLO) version 2 architecture and a squeezenet architecture. The yolo architecture is a real time object detection system. Here it used for identify and detect faces in real time. These images are captured by using anchor boxes for accuracy. The second architecture is squeezenet and is used for gender classification and age estimation. It provides significant, accurate object detection and extracts high-level features that help to achieve tremendous performance to classify the image and detecting objects. Both the architectures provide accurate result than other methods with the large no of hidden layers and cross validation in the neural network.

**CHAPTER - 4:**

**FEASIBILITY STUDY**

**4. FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**4.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**4.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**4.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**CHAPTER - 5:**

**SYSTEM REQUIREMENTS**

**5. SYSTEM REQUIREMENTS**

**5.1 HARDWARE REQUIREMENTS:**

* System : Pentium Dual Core.
* Hard Disk : 120 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 1 GB

**5.2 SOFTWARE REQUIREMENTS:**

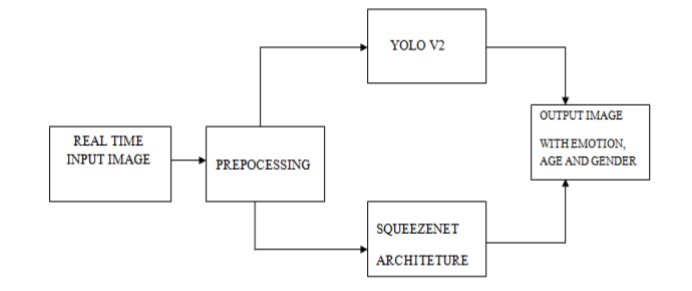
* Operating system : Windows 10
* Coding Language : python
* Tool : PyCharm
* Database : MYSQL
* Server : Flask

**CHAPTER - 6:**

**SYSTEM SESIGN**

**6. SYSTEM DESIGN**

**6.1 SYSTEM ARCHITECTURE:**

****

**6.2 DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

**User**

**Check**

**Unauthorized user**

**Yes NO**

**Upload Facial Emotional Dataset**

**Preprocess Dataset**

**Train CNN Algorithm with YOLO Faces**

**Accuracy Comparison Graph**

**Predict Facial Expression**

**End process**

**6.3 UML DIAGRAMS:**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

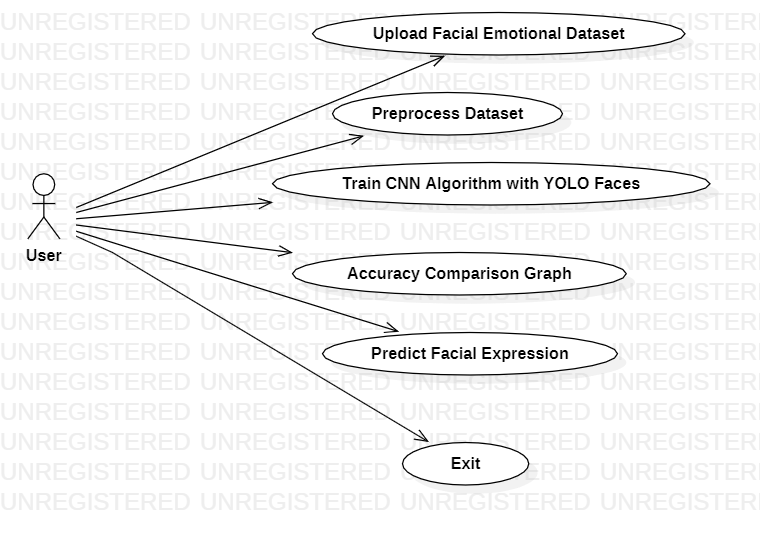
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Integrate best practices.

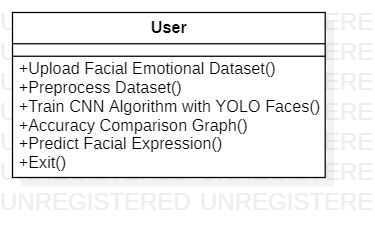
**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can bedepicted.



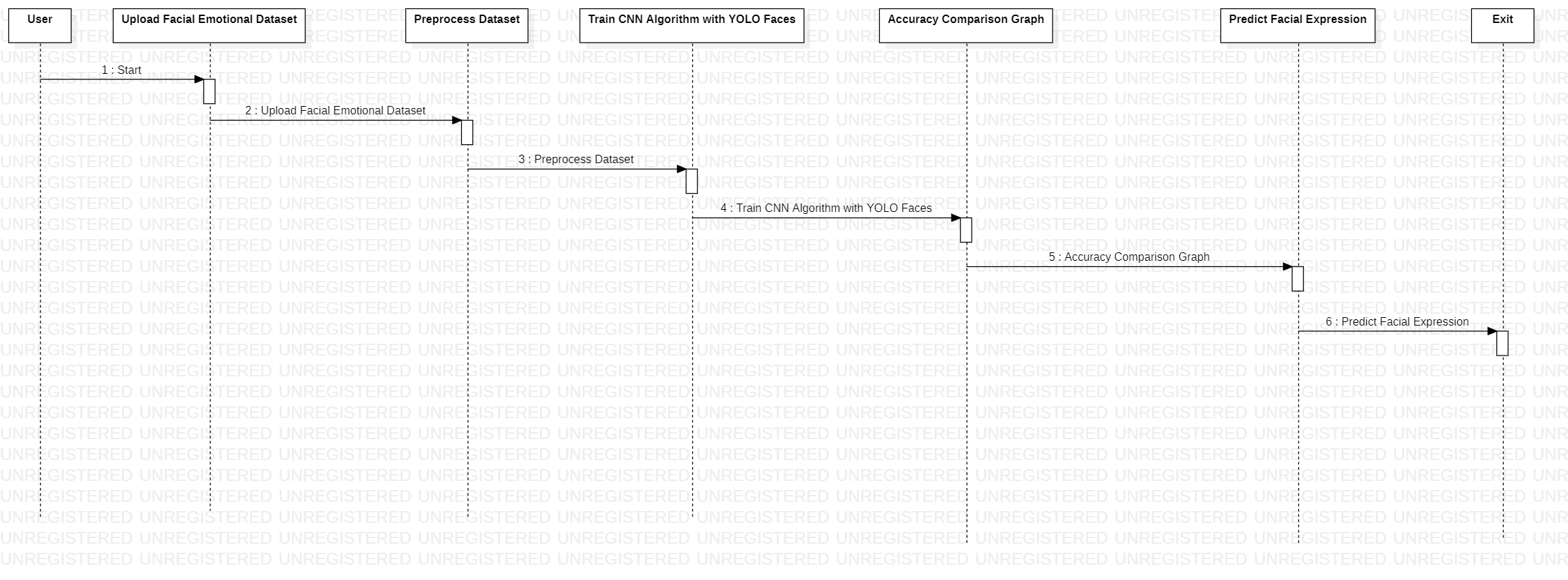
**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

****

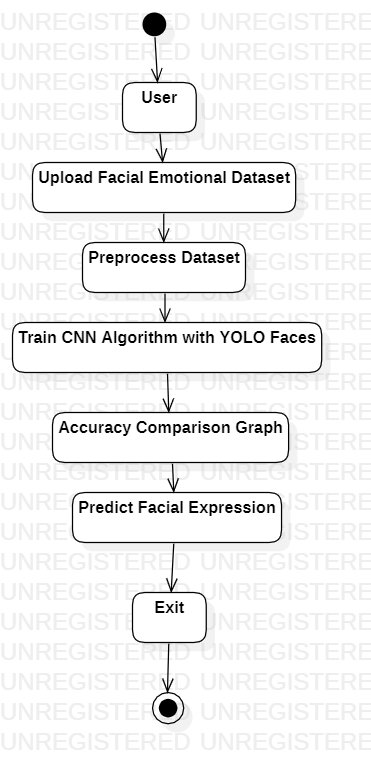
**SEQUENCE DIAGRAM:**

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. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

**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.



**CHAPTER - 7:**

**IMPLEMENTATION**

**7. IMPLEMENTATION**

**7.1 MODULES:**

* **Upload Facial Emotional Dataset**
* **Preprocess Dataset**
* **Train CNN Algorithm with YOLO Faces**
* **Accuracy Comparison Graph**
* **Predict Facial Expression**

**7.2 SAMPLE CODE:**

**Refer the code in the project**

### CHAPTER – 8

### SYSTEM TESTING

### 8. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TYPES OF TESTS**

**Unit testing:**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test:**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**8.1 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach:**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives:**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# 8.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

**8.3 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

### CHAPTER – 9:

### INPUT DESIGN AND OUTPUT DESIGN

**9. INPUT DESIGN AND OUTPUT DESIGN**

**9.1 INPUT DESIGN:**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processingcan be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES:**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2.It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**9.2 OUTPUT DESIGN:**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**CHAPTER – 10:**

**SCREENSHOTS**

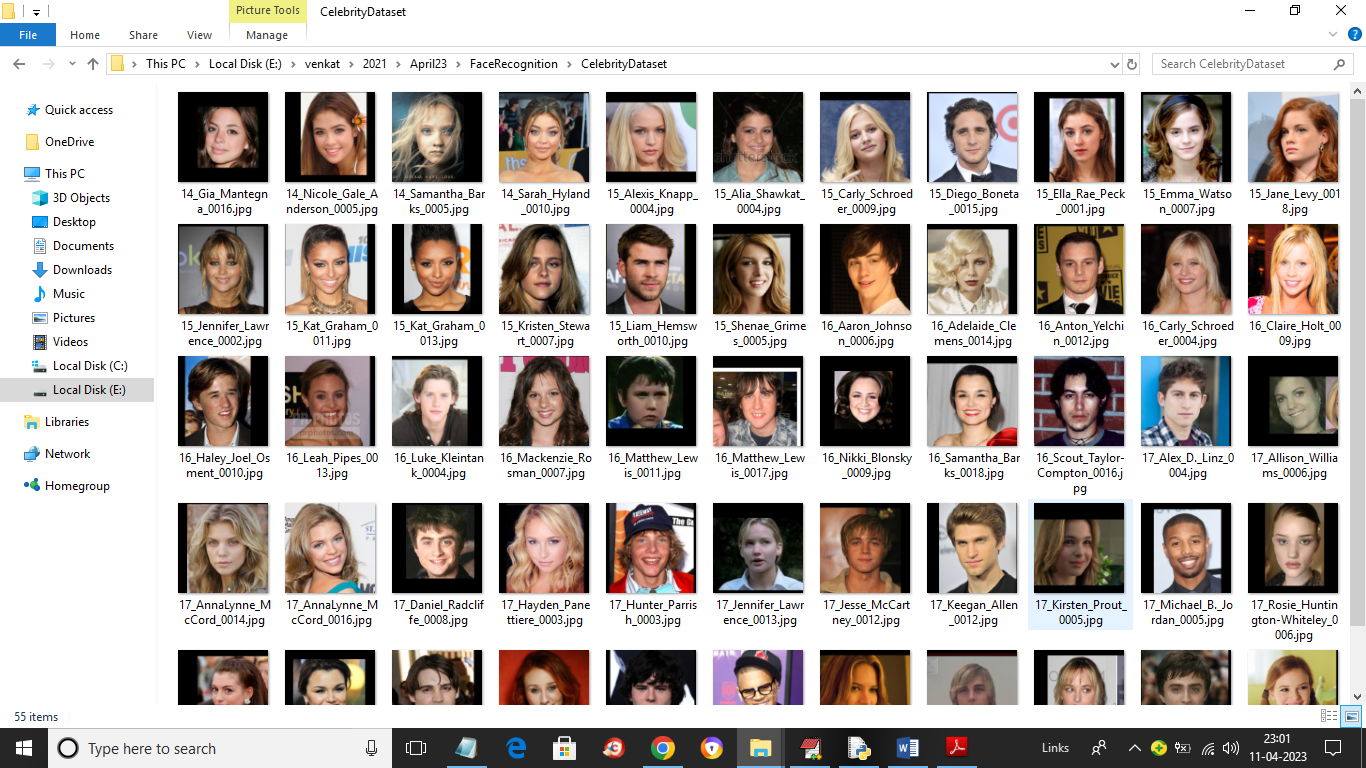
**10. SCREENSHOTS**

Face Detection & Recognition in Organic Video: A Comparative Study for Sport Celebrities Database

In this paper author is giving comparative study on face detection and recognition by using different algorithms such as HaarCascade Classifier and MMOD (Max-Margin object detection) for face detection. LSBH (Local binary pattern histogram), and CNN-based Pruned ResNet34 for face recognition.

Author is giving comparative study on HaarCascade and ResNet34 and then giving comparative study on MMOD and LSBH. In both detection and recognition algorithms HaarCascade and ResNet34 is giving better performance so we are developing this algorithms.

To train algorithms author has used sports celebrities organic video images dataset and Celebrity dataset but organic video dataset is not available on internet so we have used below celebrity dataset.

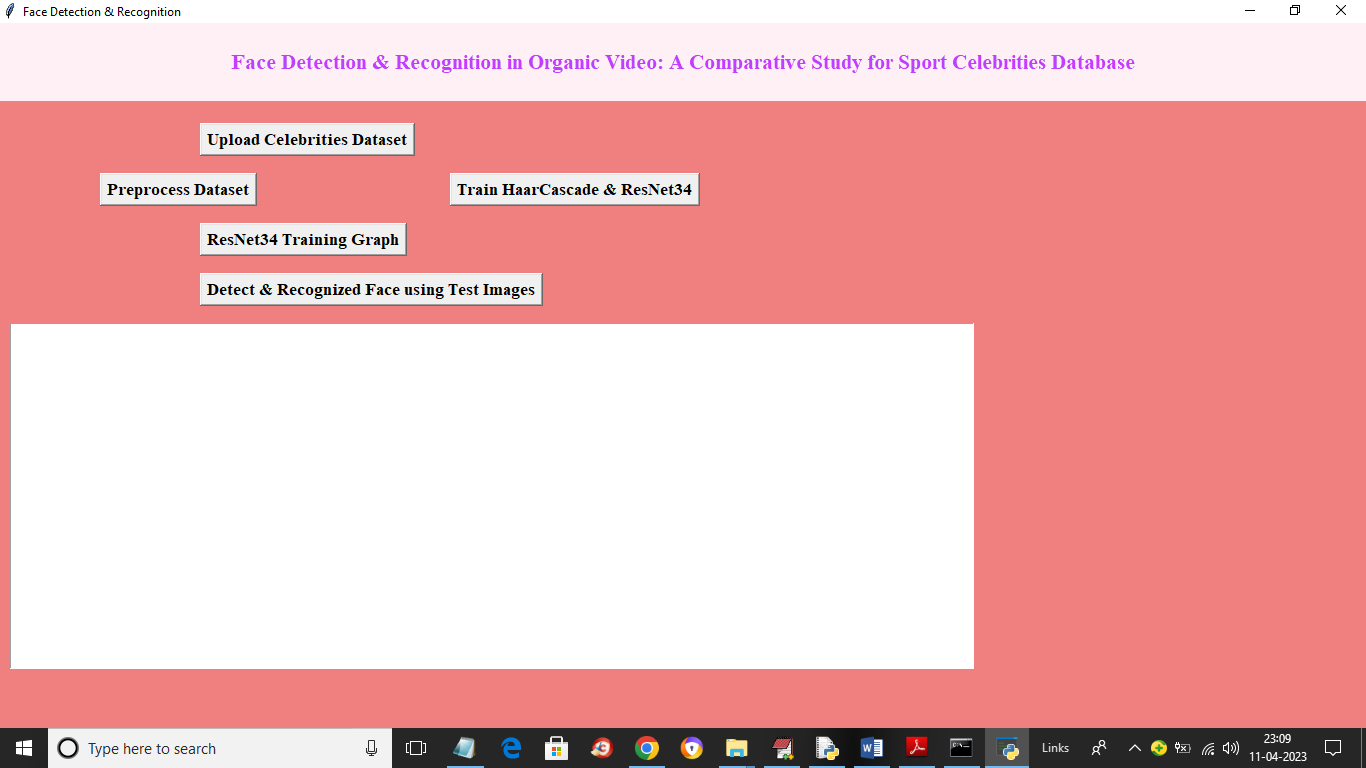


So by using above celebrity dataset we will train ResNet34 for face recognition. To implement this project we have designed following modules

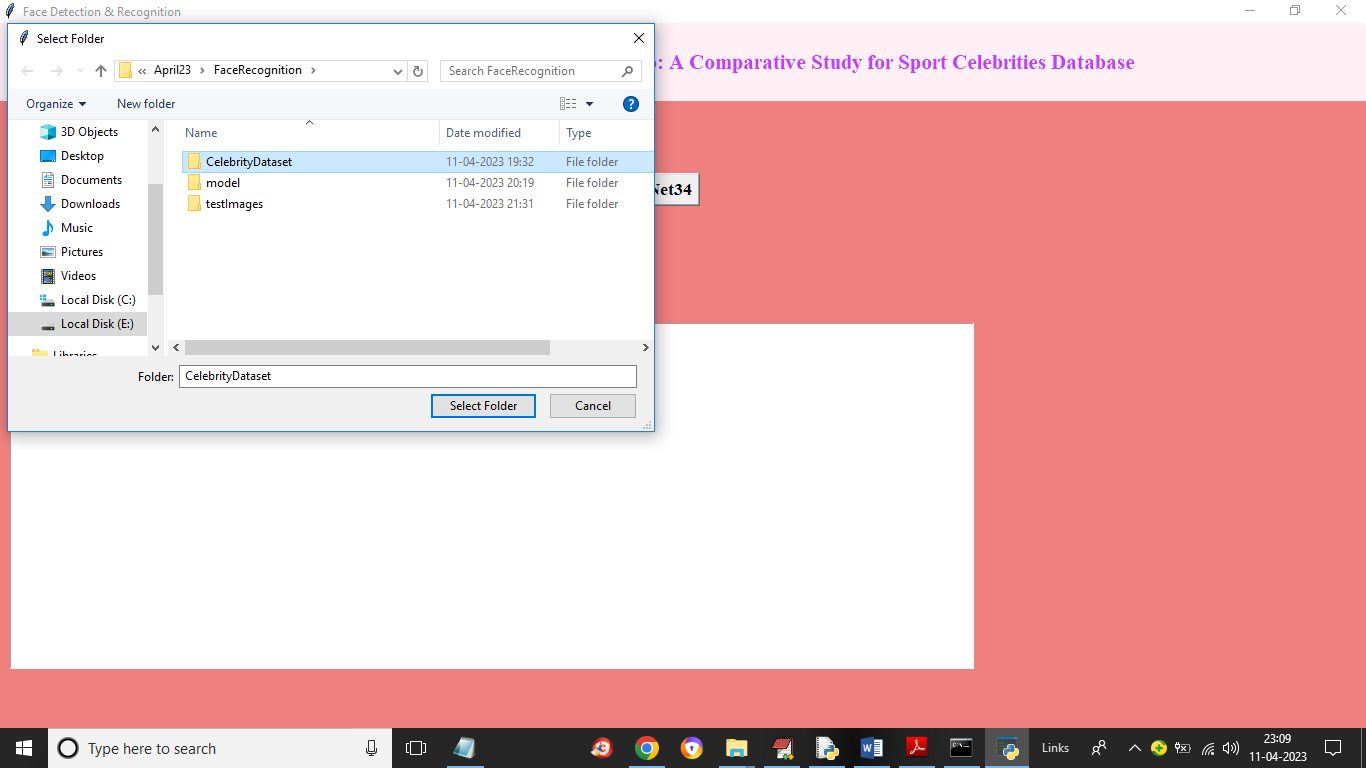
1. Upload Celebrities Dataset: using this module we will upload dataset to application
2. Preprocess Dataset: using this module we will read all images and then apply HaarCascade classifier to detect faces, normalize and shuffle faces and then split entire dataset into train and test where application using 80% dataset for training and 20% for testing
3. Train HaarCascade & ResNet34: HaarCascade face features will be input to ResNet34 to train a model which can be used to recognized faces. 80% training images will be input to ResNet34 to train a model and then apply 20% test images on trained model to calculate prediction accuracy.
4. ResNet34 Training Graph: using this module we will plot ResNet34 training accuracy and loss graph
5. Detect & Recognized Face using Test Images: using this module we will upload test image and then detect face using HaarCascade and then apply ResNet34 for face recognition.

**SCREEN SHOTS**

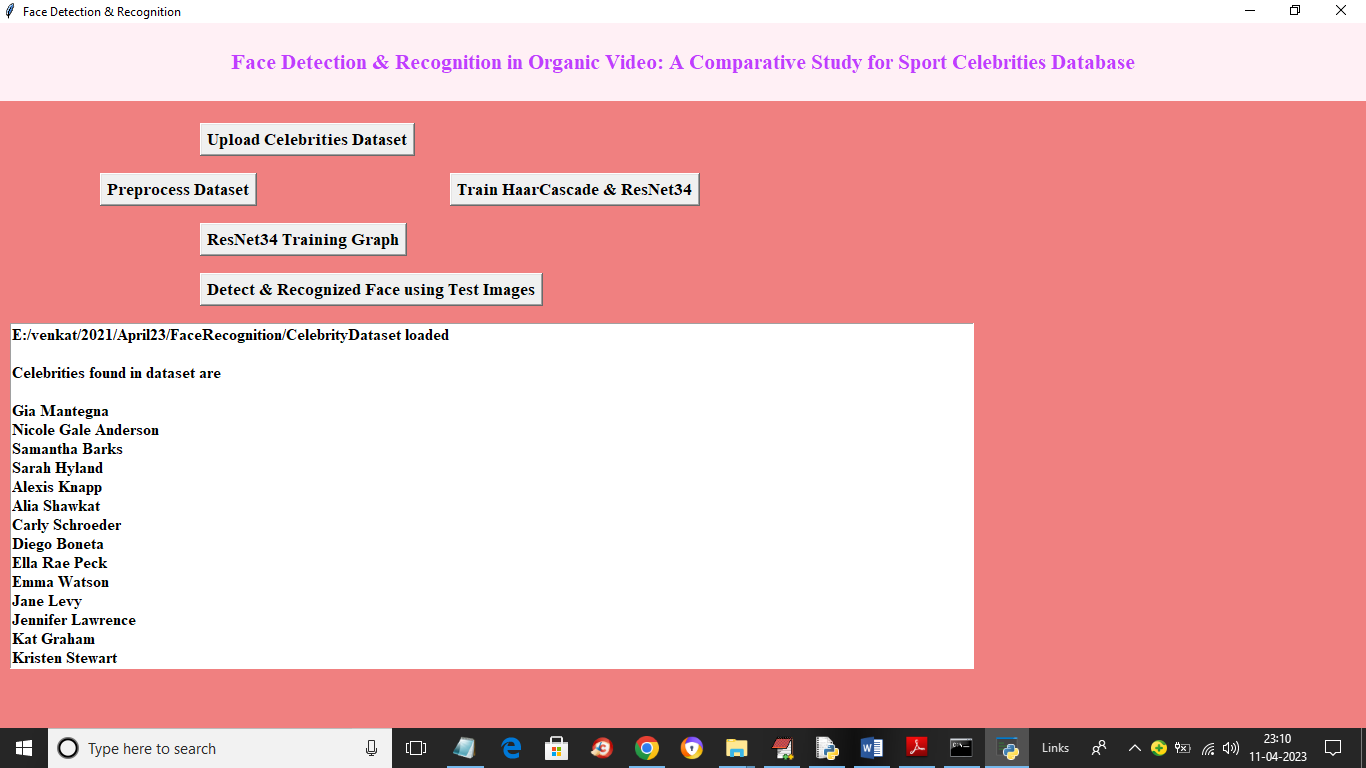
To run project double click on run.bat file to get below screen



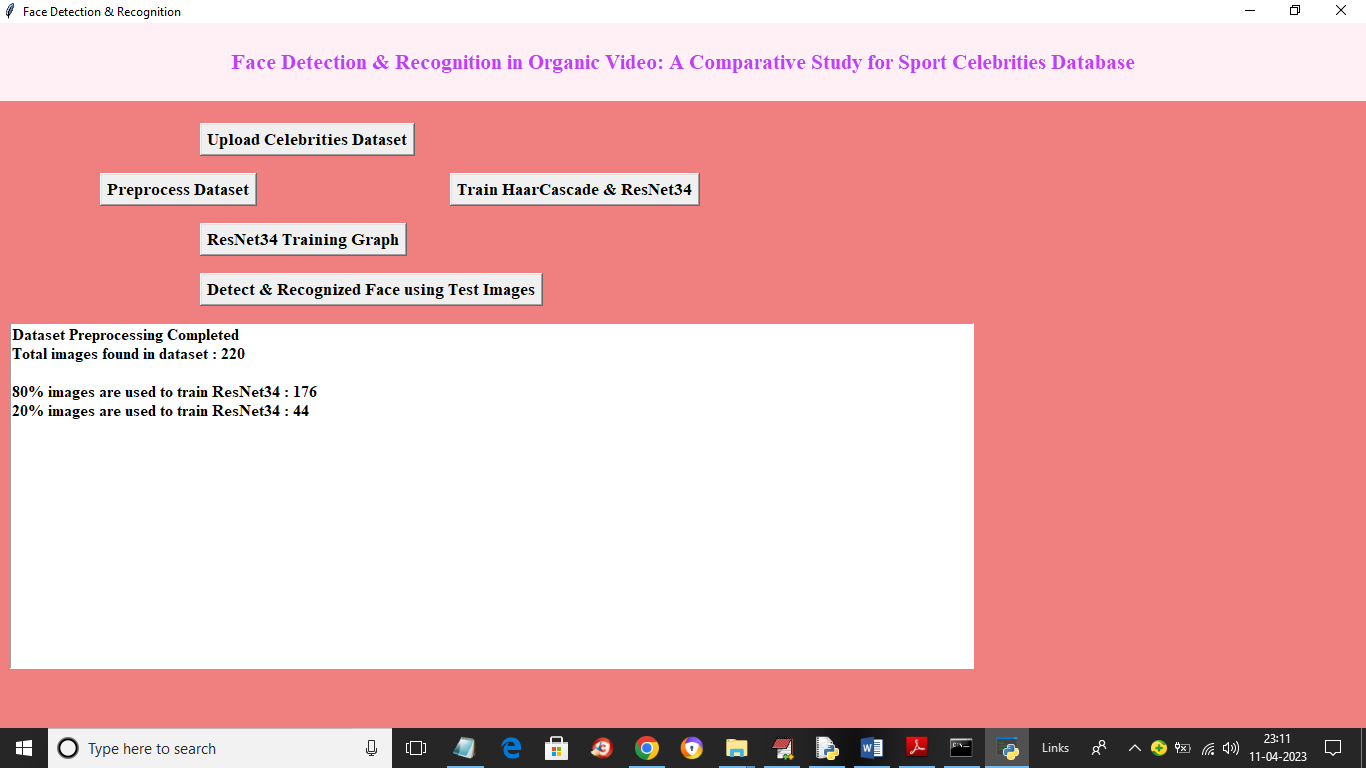
In above screen click on ‘Upload Celebrities Dataset’ button to upload dataset and get below screen



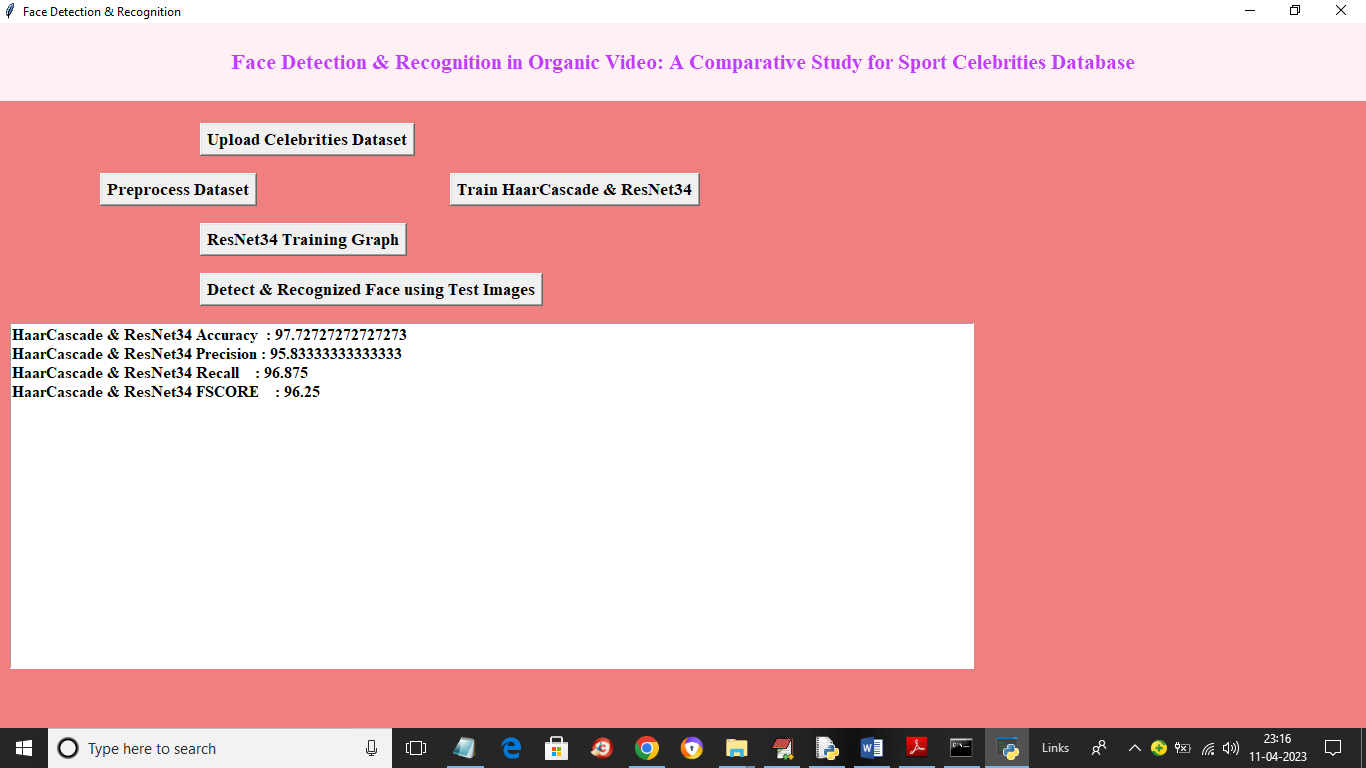
In above screen selecting and uploading ‘Celebrities Dataset’ folder and then click on ‘Select Folder’ button to load dataset and get below output



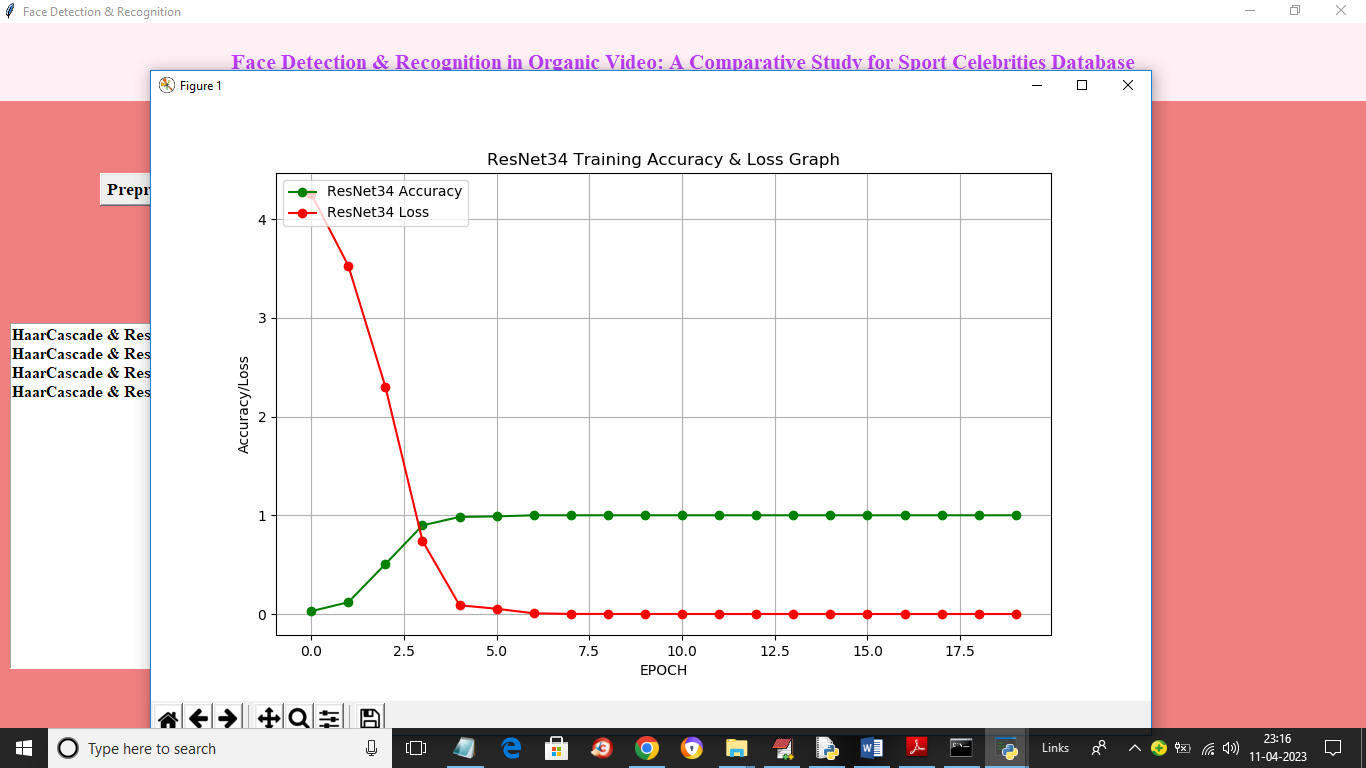
In above screen dataset loaded and we can see the names of loaded celebrity images and now click on ‘Preprocess Dataset’ button to process images like normalizing, shuffling, detecting faces and splitting it to train and test and get below output



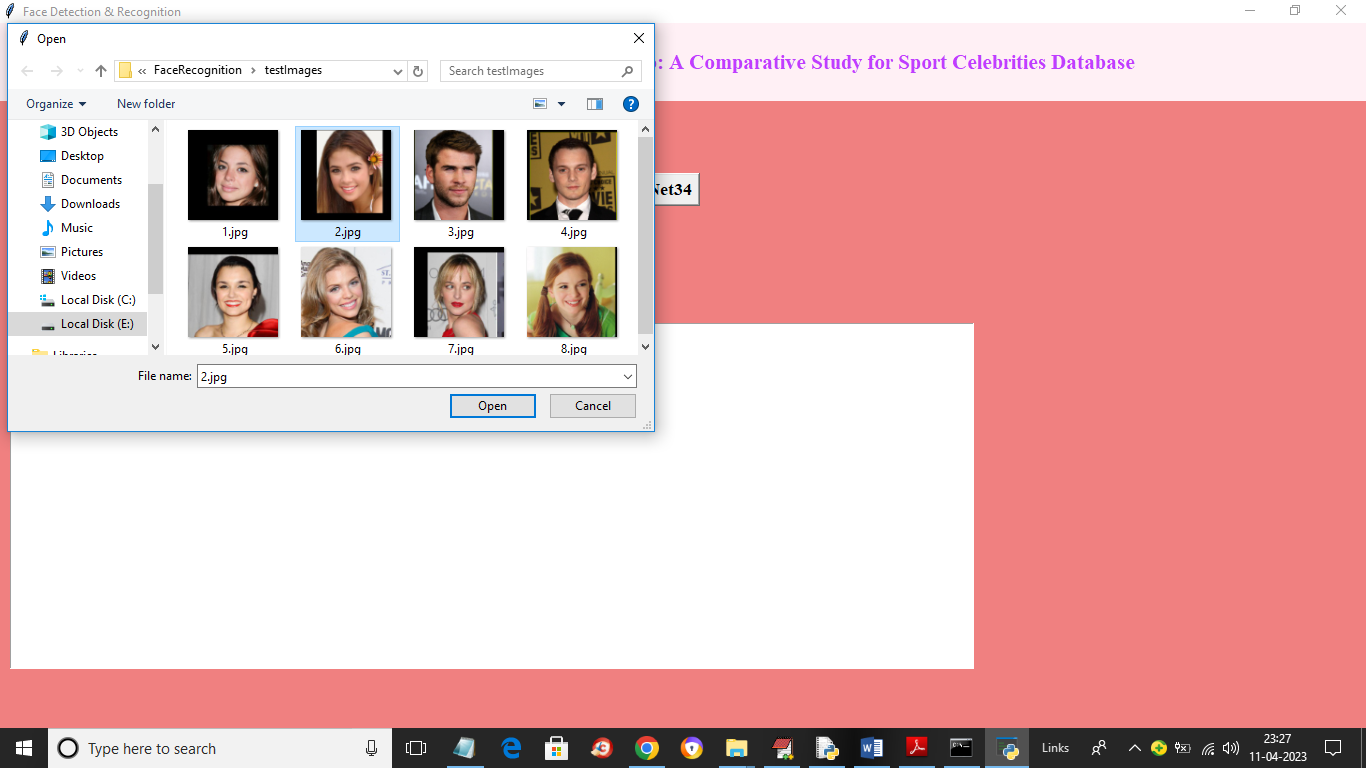
In above screen dataset processing completed and now we can see total images found in dataset and then we can see number of images using for training and testing. Now click on ‘Train Haar Cascade & ResNet34’ button to train model and get below output



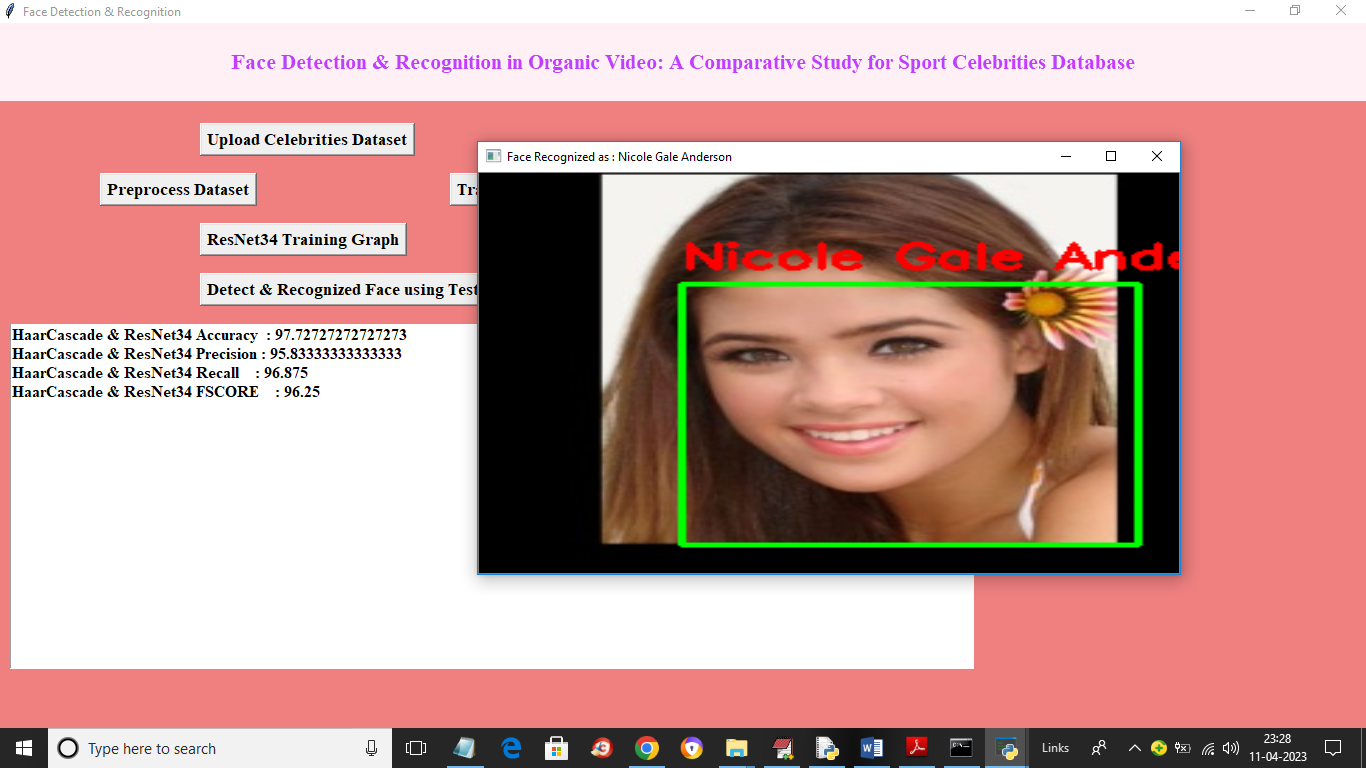
In above screen ResNet34 model trained and we got its accuracy on test data is 97% and now click on ‘ResNet34 Training Graph’ button to get below graph



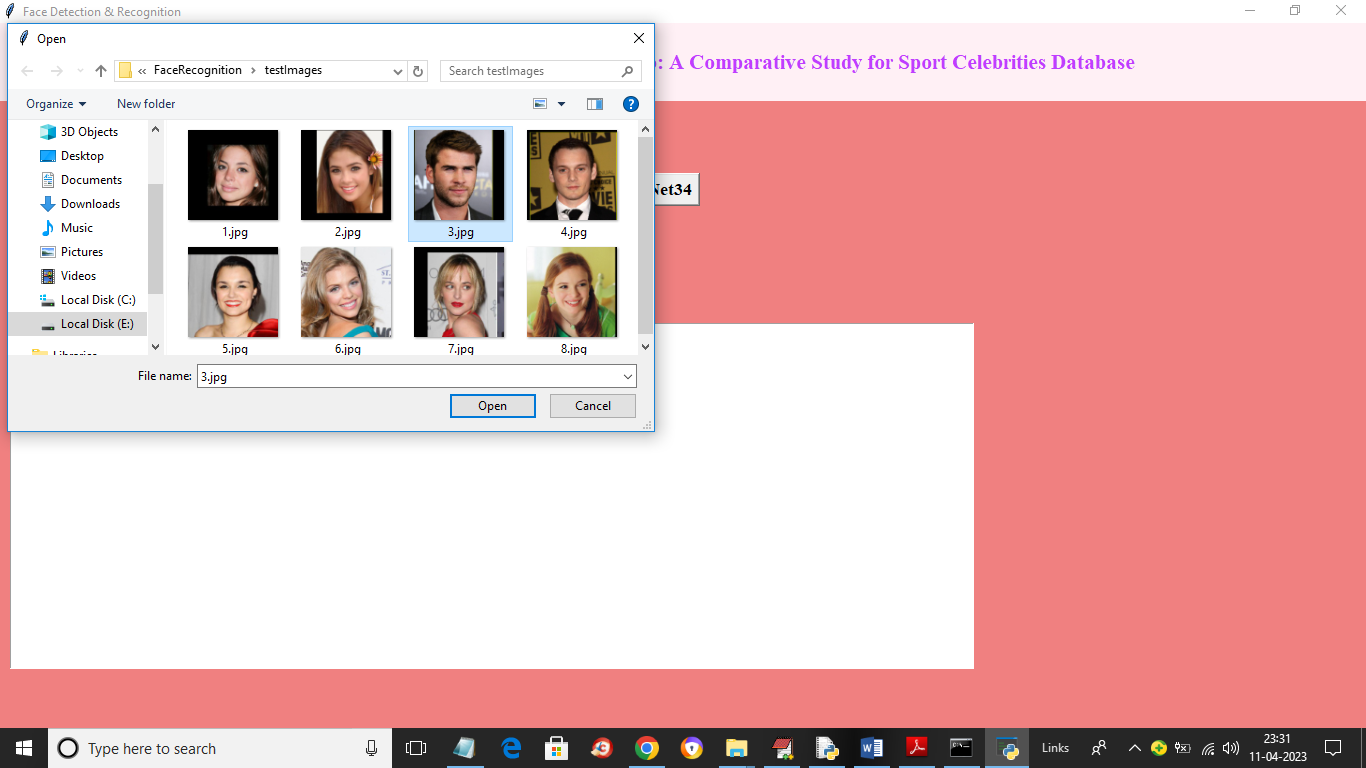
In above graph x-axis represents training epoch and y-axis represents training accuracy and loss. In above graph red line represents loss and blue line represents accuracy and with each increasing epoch accuracy got increase and reached to 1 and loss got decrease and reached to 0. Now click on ‘Detect & Recognized Face using Test Images’ button to upload image and recognize faces

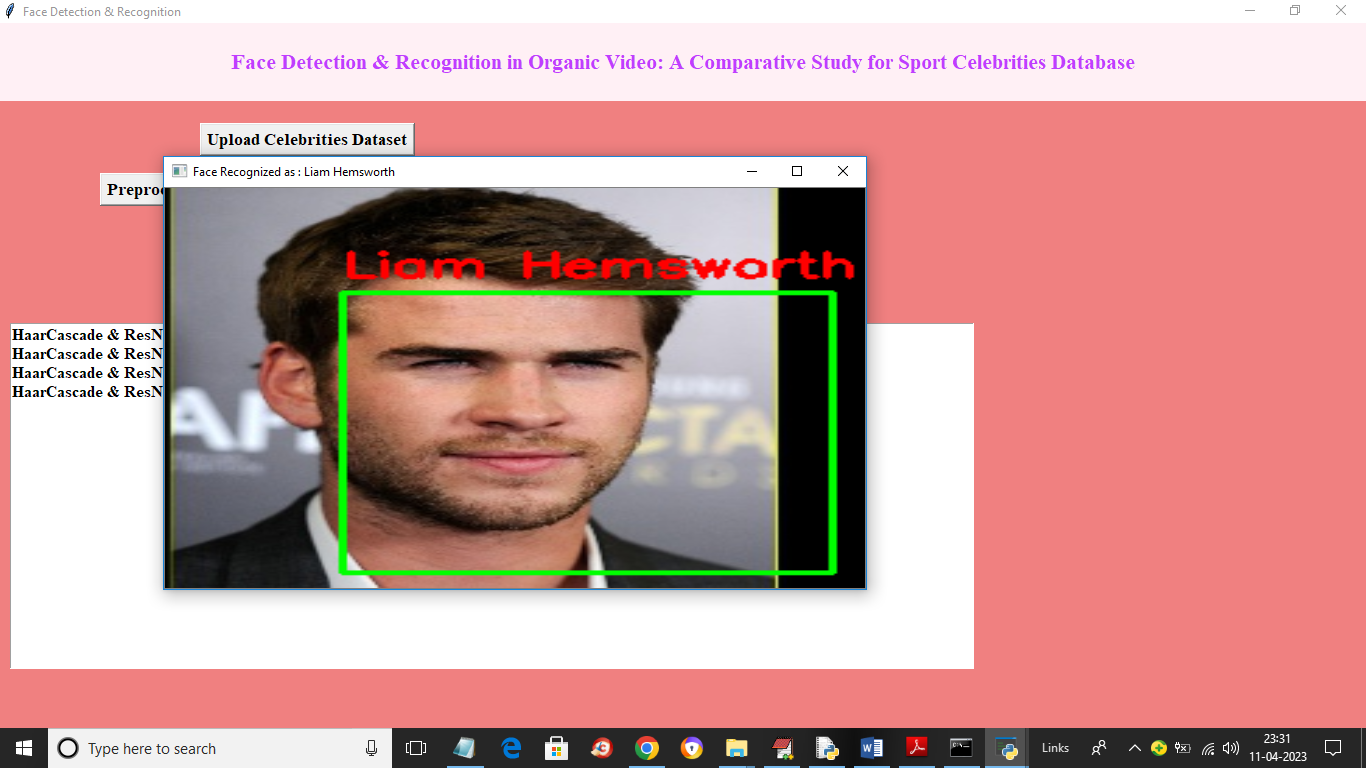


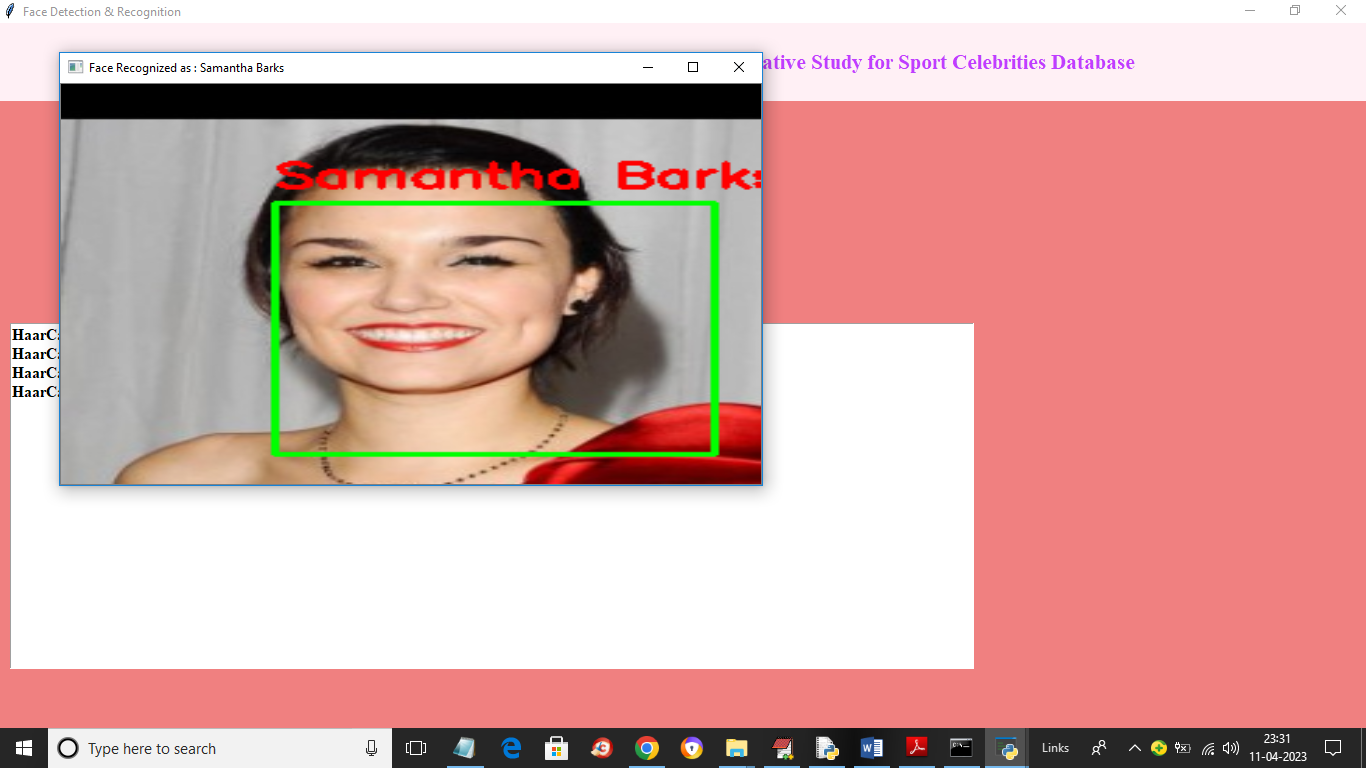
In above screen selecting and uploading ‘2.jpg’ and then click on ‘Open’ button to load image and get below output



In above screen with green line face is detected using HaarCascade and then face recognize as ‘Nicole Gale’ using ResNet34. Similarly you can upload and test other images







**CHAPTER – 11:**

**FUTURE SCOPE**

**11. FUTURE SCOPE:**

In future, this work shall be extended, and order to extract the important features from images by combining Local Binary Pattern (LBP) and HOG operator using Deep Learning models.

**CHAPTER – 12:**

**CONCLUSION**

**12. CONCLUSION**

The use of machines in society has increased widely in the last decades. Nowadays, machines are used in many different industries. As their exposure with human’s increase, the interaction also has to become smoother and more natural. In order to achieve this, machines have to be provided with a capability that let them understand the surrounding environment. Specially, the intentions of a human being. Emotion recognition is still a difficult and a complex problem in computer science because every expression is a mix of emotions. Here proposed an efficient real time facial expression recognition system with the combination of two algorithms such as yolo version 2 and squeezenet architecture based on deep neural networks for more accurate and efficient facial expression recognition. The future scope can be an action that is done upon recognition of the emotions. If get a sad emotion, can have the systems plays a song or tells a joke or send his/her best friend a message. This can be the next step of AI where the system can understand, comprehend the user’s feelings and emotions and react accordingly. This bridges the gap between machines and humans. We can also have an interactive keyboard where the users can just use the app and the app will then identify the emotion and convert that emotion to the emoticon of choice.

**CHAPTER – 13:**

**BIBLIOGRAPHY (OR) REFERENCES**

**13. BIBLIOGRAPHY (OR) REFERENCES**

[1] Jumani, S.Z., Ali, F., Guriro, S., Kandhro, I.A., Khan, A. and Zaidi, A., 2019. Facial Expression Recognition with Histogram of Oriented Gradients using CNN. Indian Journal of Science and Technology, 12, p.24.

[2] J. R. Barr, L. A. Cament, K. W. Bowyer, and P. J. Flynn. Active clustering with ensembles for social structure extraction. In Winter Conference on Applications of Computer Vision, pages 969–976, 2014

[3] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[4] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).R. Goh, L. Liu, X. Liu, and T. Chen. The CMU face in action (FIA) database. In International Conference on Analysis and Modelling of Faces and Gestures, pages 255–263. 2005.

[5] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In Computer Vision and Pattern Recognition, pages 529–534, 2011

[6] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B. C. Lovell. Patchbased probabilistic image quality assessment for face selection and improved video-based face recognition. In Computer Vision and Pattern Recognition Workshops, pages 74–81, 2011.

[7] B. R. Beveridge, P. J. Phillips, D. S. Bolme, B. A. Draper, G. H. Givens, Y. M. Lui, M. N. Teli, H. Zhang, W. T. Scruggs, K. W. Bowyer, P. J. Flynn, and S. Cheng. The challenge of face recognition from digital point-and-shoot cameras. In Biometrics: Theory Applications and Systems, pages 1–8, 2013

[8] N. D. Kalka, B. Maze, J. A. Duncan, K. A. O Connor, S. Elliott, K. Hebert, J. Bryan, and A. K. Jain. IJB–S: IARPA Janus Surveillance Video Benchmark. In IEEE International Conference on Biometrics: Theory, Applications, and Systems, 2018.

[9] M. Singh, S. Nagpal, N. Gupta, S. Gupta, S. Ghosh, R. Singh, and M. Vatsa. Cross-spectral cross-resolution video database for face recognition. In IEEE International Conference on Biometrics Theory, Applications and Systems, 2016 [10] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 2879–2886.

[11] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, “Joint cascade face detection and alignment,” in Proc. Eur. Conf. Comput. Vis., 2014, vol. 8694, pp. 109–122. [12] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 2650– 2658

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[15] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1915–1929. [16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

[17] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 580–587.

[18] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, International Conference on Learning Representations.

[19] DESivaram, M., Porkodi, V., Mohammed, A.S. and Manikandan, V., 2019. Detection Of Accurate Facial Detection Using Hybrid Deep Convolutional Recurrent Neural Network. ICTACT Journal on Soft Computing, 9(2).

[20] Sivaram, M., Porkodi, V., Mohammed, A.S. and Manikandan, V., 2019. Detection of Accurate Facial Detection Using Hybrid Deep Convolutional Recurrent Neural Network. ICTACT Journal on Soft Computing, 9(2).

[21] Shafiee, M.J., Chywl, B., Li, F. and Wong, A., 2017. Fast YOLO: a fast you only look once system for real-time embedded object detection in video. arXiv preprint arXiv:1709.05943.

[22] Ravidas, S. and Ansari, M.A., 2019. An Efficient Scheme of Deep Convolution Neural Network for Multi View Face Detection. International Journal of Intelligent Systems and Applications, 11(3), p.53.

[23] Gupta, S., Gupta, N., Ghosh, S., Singh, M., Nagpal, S., Vatsa, M. and Singh, R., 2019.