XMOJI (EXPRESSION TO EMOJI)

A Mini project report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
In
Computer Science and Engineering

Submitted By

M. SARATH KUMAR	(121710306049)
K. YASWANTH	(121710306029)
G.SATYA SAI UDAY	(121710306020)
S.HARSHA	(121710306052)

Under the esteemed guidance of

Dr M. Padmaja
Assistant Professor

Under the supervision of

Mrs. P Mansa Devi
Prof. T SitaMahalakshmi

Department of Computer Science and Engineering

GITAM(Deemed to be University)

Visakhapatnam

January 2021

DEPARTMENT OF COMUTER SCIENCE AND ENGNIEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM(Deemed to be University)

VISAKHAPATNAM

DECLARATION

We hereby declare that the project review entitled "**XMOJI(EXPRESSION TO EMOJI)**" is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University, VISAKHAPATNAM) submitted in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date: 13 April 2020

Registration No.	**Names**	**Signature**
121710306049	M.SARATH KUMAR	
121710306029	K.YASWANTH	
121710306020	G.UDAY	
121710306052	S.HARSHA	

DEPARTMENT OF COMUTER SCIENCE AND ENGNIEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM(Deemed to be University)

VISAKHAPATNAM

CERTIFICATE

This is to certify that the mini project documentation report entitled "**XMOJI(EXPRESSION TO EMOJI)**" is bonified record of work carried out by **M.SARATH KUMAR(121710306049), K.YASWANTH(121710306029), G.UDAY(121710306020), S.HARSHA(121710306052)** students submitted in fulfilment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide	**Supervisors**
Dr M. Padmaja	**Dr. K. Thammi Reddy**
Assistant Professor	**Professor**

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to our final year project guide, **Dr M. PADMAJA**, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report.

Furthermore, we would also like to acknowledge with much appreciation the crucial role of our project reviewers **Mrs. P.MANASA DEVI, Prof T.SITAMAHALAKSHMI** mam who have been a lot of help and support during our project phase. I have to appreciate the guidance given by other supervisors and teammates as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comments and advices.

We wish to express our gratitude to **Dr. K THAMMI REDDY** sir, Head of the Department, Computer Science and Engineering, GITAM (Deemed to be University) for providing a great support for us in completing our project by arranging the trainees, faculty needed to complete our project and for giving us the opportunity for doing this project.

ABSTRACT

The main idea is to capture the human face through live web proctoring and recognize the facial expressions stimulated in the face and describe the age, gender and create and display an emoji according to the expression. The main goal of our project is that emoji's and avatars are ways to indicate non-verbal cues or part of communication. These have become an essential part of online chatting, product review, brand emotion, and many more. It also leads to increasing data-science research dedicated to emoji driven storytelling. Social media platforms like snap chat have some very cool features such as lenses and emojis (bitmoji) and Apple Company has also launched its animated characters Animoji and Memoji. In this article, you will learn how to develop emoji that mirrors your face in real-time. Real-time emoji is an animated version of popular emoji characters. They essentially mirror your facial expressions. So, as you move your face and talk, they will do the same, in realtime.

Keywords: Face Expression Detection, Gender Detection, Age Detection

TABLE OF CONTENTS

1. Introduction……………………………………………………………….…7

2. Literature Review…………………………………………………………..8

3. Existing Systems……………………………………………………...11

4. Problem Statement………………………………………………………..12

5. System Architecture……………………………………………….....14

6. Datasets……………………………………………………………………17

7. Requirements and Specifications..…..24

8. Programming Language……………………………………………………26

9. Feasibility Report………………………………..……………………35

10. Uml Design……………..……………………………………………….37

11.Implementation of GUI Using Flask……………………………………40

12. Test Cases……………………………………………………………….47

13. Conclusion…………………………………………………………………51

14. Future Scope………………………………………………………………52

15. References……………………………………………………………….53

1. INTRODUCTION

You expect employees to have high levels of emotional intelligence when interacting with customers. Now, thanks to advances in Deep Learning, you'll soon expect your software to do the same. Research has shown that over 90 percent of our communication can be non-verbal, but technology has struggled to keep up, and traditional code is generally bad at understanding our intonations and intentions. But emotion recognition—also called Affective Computing—is becoming accessible to more types of developers. This post will walk through the ins-and-outs of determining emotion, gender and age from data, and a few ways you can get some expression detection and running yourself.

What is facial expression detection?

Facial expression detection is the process of detecting human emotions from facial expressions. The human brain recognizes emotions automatically, and software has now been developed that can recognize emotions as well. This technology is becoming more accurate all the time, and will eventually be able to read emotions as well as our brains do. AI can detect emotions by learning what each facial expression means and applying that knowledge to the new information presented to it. Emotional artificial intelligence, or emotion AI, is a technology that is capable of reading, imitating, interpreting, and responding to human facial expressions and emotions.

Expression Detection Use Cases: TSA Screening, Audience Engagement, And More facial emotion recognition detector.

BACKGROUND

Facial expression detection is crucial in daily life in order to identify facial feeling or expression of someone. We might not perceive that several steps have actually taken in order to identify human emotions. The current method that we follow is we visually see the face of the person and then we identify his emotions and then we identify his feelings. In some cases directly we cannot meet the person to know his emotions in such case this facial expression detection is very helpful to know the emotion of a person. So, why not shift to a facial expression detection system which works on face recognition technique? To identify the emotion of a person easily.

2. LITERATURE REVIEW

LITERATURE STUDY

As per various literature surveys it is found that for implementing this project four basic steps are required to be performed.

i. Preprocessing
ii. Face registration
iii. Facial feature extraction
iv. Expression classification
v. Gender and age classification
vi. Face mask and without face mask classification
vii. Sql database and front end for deploying model using flask

Description about all these processes are given below-

Preprocessing

Preprocessing is a common name for operations with images at the lowestlevel of abstraction both input and outputs are intensity images.

Face Registration:

Face Registration is a computer technology being used in a variety ofapplications that identifies human faces in digital images. In this face registration step, faces are first located in the image using some set of landmark points called “face localization” or “face detection”. These detected faces are then geometrically normalized to match some template image in a process called “face registration”.

Facial Feature Extraction

Facial Features extraction is an important step in face recognition and is defined as the process of locating specific regions, points, landmarks, or curves/contours in a given2-D image or a 3D range image. In this feature extraction step, a numerical feature vector is generated from the resulting registered image. Common features that can be extracted are-

a. Lips
b. Eyes
c. Eyebrows
d. Nose tip

Gender and age classification

Age and gender predictions of unfiltered faces classify unconstrained real-world facial images into predefined age and gender. Significant improvements have been made in this research area due to its usefulness in intelligent real-world applications. More recently, Convolutional Neural Networks (CNNs) based methods have been extensively used for the classification task due to their excellent performance in facial analysis. In this work, we propose a novel end-to-end CNN approach, to achieve robust age group and gender classification of unfiltered real-world faces.

Face mask and without face mask classification

In the new world of corona virus, multidisciplinary efforts have been organized to slow the spread of the pandemic. The AI community has also been a part of these endeavors. In particular, developments for monitoring social distancing or identifying face masks have made-the-headlines.

Expression Classification

In the third step, of classification, the algorithm attempts to classify the given faces portraying the basic emotions.Paul Ekman (born February 15, 1934) is an American psychologist and professor emeritus at the University of California, San Francisco who is a pioneer in the study of emotions and their relation to facial expressions. He has created an "atlas of emotions" with more than ten thousand facial expression.

Deploy Machine Learning Models using Flask

In a typical machine learning and deep learning project, we usually start by defining the problem statement followed by data collection and preparation, understanding of the data, and model building, right. But, in the end, we want our model to be available for the end-users so that they can make use of it. Model Deployment is one of the last stages of any machine learning project and can be a little tricky. Here comes the role of Flask. Flask is a web application framework written in Python. It has multiple modules that make it easier for a web developer to write applications without having to worry about the details like protocol management, thread management, etc.

3. EXISTING SYSTEMS

Let us consider the messaging system, it contains messaging, voice, emotions and many more but the actual mood off the user is not detected. Even in medical fields like psychology or psychiatry, there is no virtual emotion. There could also be virtual emotion detection in these required fields for detecting a person’s emotion or expressions in real time.

Drawbacks of existing system:-

- There is no virtual emotion detection particularly in the existing systems/fields.
- The emotion detection makes the work easier and due to absence of this feature the process of messaging system becomes complicated and more normal to use and gets bored.
- These field require emotion detection as major part as they have to work or generate the results based on the emotions of person which will be the major part of their working according to the emotion such as generating texts, etc…

Proposed System

Our proposed system is facial expression detector. Facial Expression or Emotion detector is used to know or detect whether the person is happy, sad, and angry and so on only through his or her face. When this gets installed in a system, it can predict the expression or emotion of a person in real time. This particular system is easy to use and gives good results in real time.

4. PROBLEM STATEMENT

Human facial expressions can be easily classified into 7 basic emotions: happy, sad, surprise, fear, anger, disgust, and neutral, our facial emotions are expressed through activation of specific sets of facial muscles.In addition to those 7 facial expressions we also want to add a face mask detection, the face mask which is a common thing in our daily life due to covid-19 disease.These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind. Through facial expression detection, we are able to measure the effects that content and services have on the audience/users through an easy and low-cost procedure.

For example, retailers may use these metrics to evaluate customer interest. Healthcare providers can provide better service by using additional information about patients' emotional state during treatment. Entertainment producers can monitor audience engagement in events to consistently create desired content. Humans are well-trained in reading the emotions of others, in fact, at just 14 months old, babies can already tell the difference between happy and sad. But can computers do a better job than us in accessing emotional states? To answer the question, we designed a deep learning neural network that gives machines the ability to make inferences about our emotional states, gender and age which also gives a related emoji on based upon your face expression, age and gender. In other words, we give them eyes to see what we can see.We named our project as Xmoji (expression to emoji).

Solution to the problem statement

The objective of this project is to develop facial eexpression detection system.

Expected achievements in order to fulfill the objectives are:

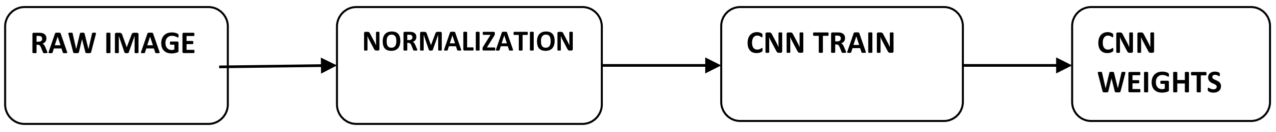
1. To collect the data sets for 7 face expressions and face mask, 2 genders and 6 age groups.
2. To train the model from data set using convolution neural networks.
3. To predict and recognize the face detected.
4. Identify the expressions, gender, and age.
5. Deliver the output on the screen in words and related emoji based upon the age, gender, facial expression. (i.e.; emotion of the person that has been detected).

6. To collect the emojis of different expressions, genders and age groups.
7. To deploy the facial expression detection into web page using flask and host.
8. To add the user authentication, user verification to use the Xmoji.

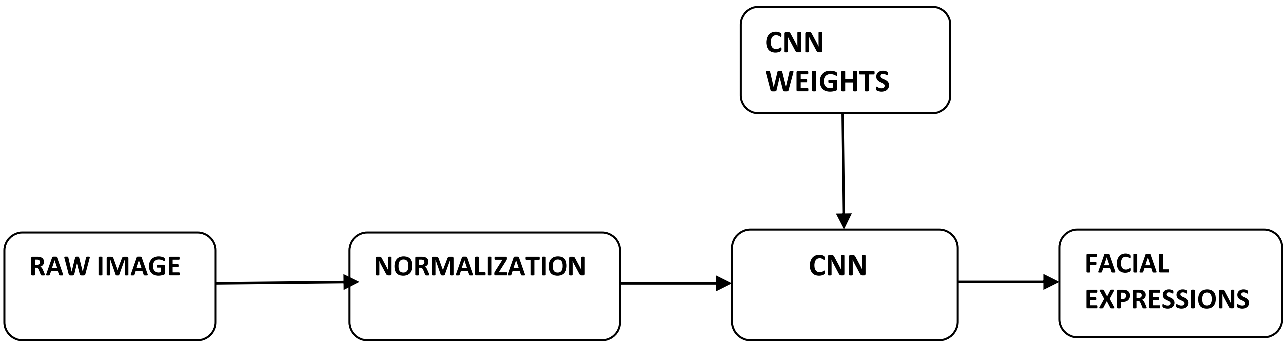
5. SYSTEM ARCHITECTURE

The facial expression recognition system is implemented using Convolutional Neural Networks.

The block diagram of the system is shown below:-



(a): TRAINING PHASE



(b): TESTING PHASE

During training, the system received a training data comprising grayscale images of faces with their respective expression label and learns a set of weights for the network. The training step took as input an image with a face. Thereafter, intensity normalization is applied to the image. The normalized images are used to train the Convolutional Network. To ensure that the training performance is not affected by the order of presentation of the examples, validation dataset is used to choose the final best set of weights out of a set of trainings performed with samples presented in different orders. The output of the training step is a set of weights that achieve the best result with the training data.

During test, the system received a grayscale image of a face from test dataset, and output the predicted expression by using the final network weights learned during training. Its output is a single number that represents one of the seven basic expressions.

ARCHITECTURE OF CNN:-

A typical architecture of a convolutional neural network contains an input layer, some convolutional layers, some fully-connected layers, and an output layer. It has 6 layers without considering input and output. The architecture of the Convolution Neural Network used in the project.

1. Input Layer

The input layer has pre-determined, fixed dimensions, so the image must be -processed before it can be fed into the layer. Normalized gray scale images of size 48 X 48 pixels from Kaggle dataset are used for training, validation and testing. For testing propose laptop webcam images are also used, in which face is detected and cropped using OpenCVHaar Cascade Classifier and normalized.

2. Convolution and Pooling (ConvPool) Layers

Convolution and pooling is done based on batch processing. Each batch has N images and CNN filter weights are updated on those batches. Each convolution layer takes image batch input of four dimension N x Color-Channel x width x height. Feature map or filters for convolution are also four dimensional (Number of feature maps in, number of feature maps out, filter width, filter height). In each convolution layer, fourdimensional convolution is calculated between image batch and feature maps. After convolution only parameter that changes is image width and height.

New image width = old image width – filter width + 1
New image height = old image height – filter height + 1

After each convolution layer down sampling / sub sampling is done fordimensionality reduction. This process is called Pooling. Max pooling and Average Pooling are two famous pooling methods. In this project max pooling is done after convolution.

Two convolution layer and pooling layer are used in the architecture. At first convolution layer size of input image batch is Nx1x48x48. Here, size of image batch is

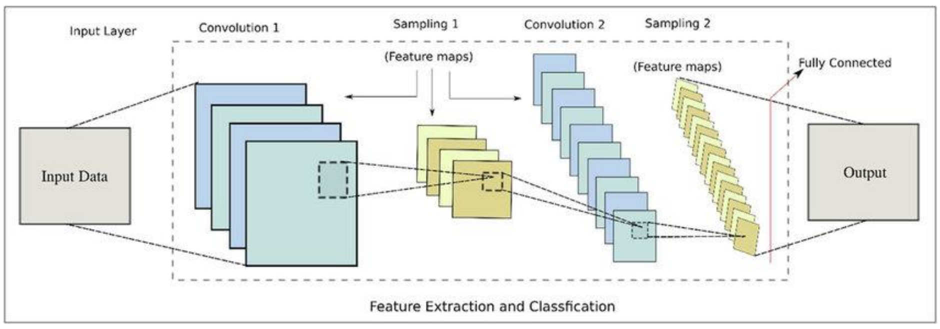
N, number of color channel is 1 and both image height and width are 48 pixel. Convolution with feature map of 1x20x5x5 results image batch is of size Nx20x44x44. After convolution pooling is done with pool size of 2x2, which results image batch of size Nx20x22x22. This is followed by second convolution layer with feature map of 20x20x5x5, which results image batch of size Nx20x18x18. This is followed by pooling layer with pool size 2x2, which results image batch of size Nx20x9x9.

3. Fully Connected Layer

This layer is inspired by the way neurons transmit signals through the brain. It takes a large number of input features and transforms features through layers connected with trainable weights. Two hidden layers of size 500 and 300 unit are used in fully-connected layer. The weights of these layers are trained by forward propagation of training data then backward propagation of its errors. Back propagation starts from evaluating the difference between prediction and true value, and back calculates the weight adjustment needed to every layer before. We can control the training speed and the complexity of the architecture by tuning the hyper-parameters, such as learning rate and network density. Hyper-parameters for this layer include learning rate, momentum, regularization parameter, and decay.

4. Output Layer

Output from the second hidden layer is connected to output layer having seven distinct classes. Using Softmax activation function, output is obtained using the probabilities for each of the seven classes. The class with the highest probability is the predicted class.



6. DATATSETS

Information about Datasets

- Number of datasets used- 3(Expression data set,Gender dataset,Mask dataset)
- Expression dataset- 43,192 images
- Gender dataset- 2307 images
- Face mask dataset - 1915 images
- Emojis dataset - 224 images

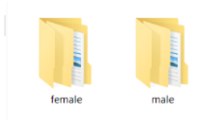
Source: Kaggle,Google.

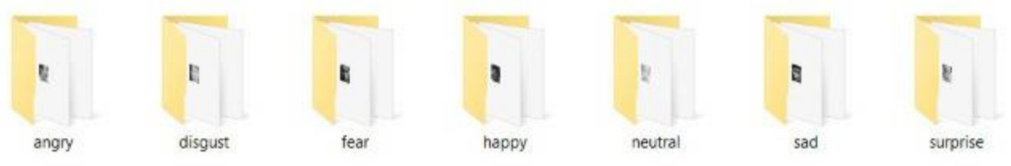
The dataset from a Kaggle is used for the training and testing. It comprises pre-cropped, 48-by-48-pixel grayscale images of faces each labeled with one of the 7 emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. Dataset has training set of 35,887 facial images with facial expression labels. The dataset has class imbalance issue, since some classes have large number of examples while some has few. The dataset is balanced using oversampling, by increasing numbers in minority classes. The balanced dataset of expressions contains 43,192 images, from which 34,760 images are used for training, 8432 images are used for testing.

The dataset of mask images contains 1915 images and also the dataset for gender detection is collected using web scraping and contains a total of 2307 images with 1173 male images and 1134 female images.

Emojis Dataset

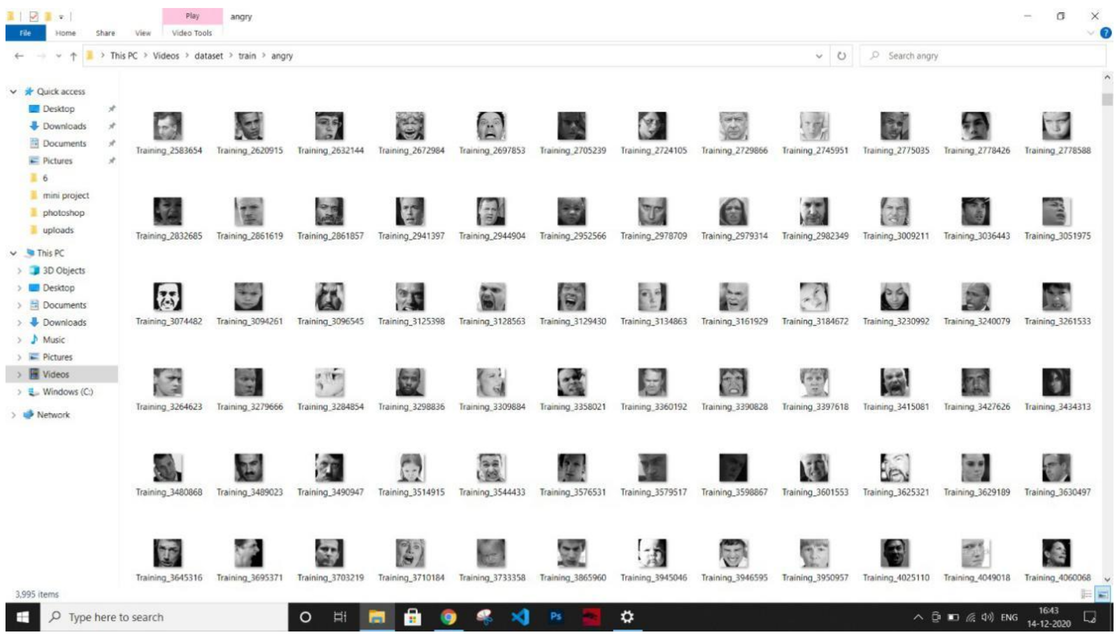
This dataset is collected to give the output for a particular expression,gender and age.



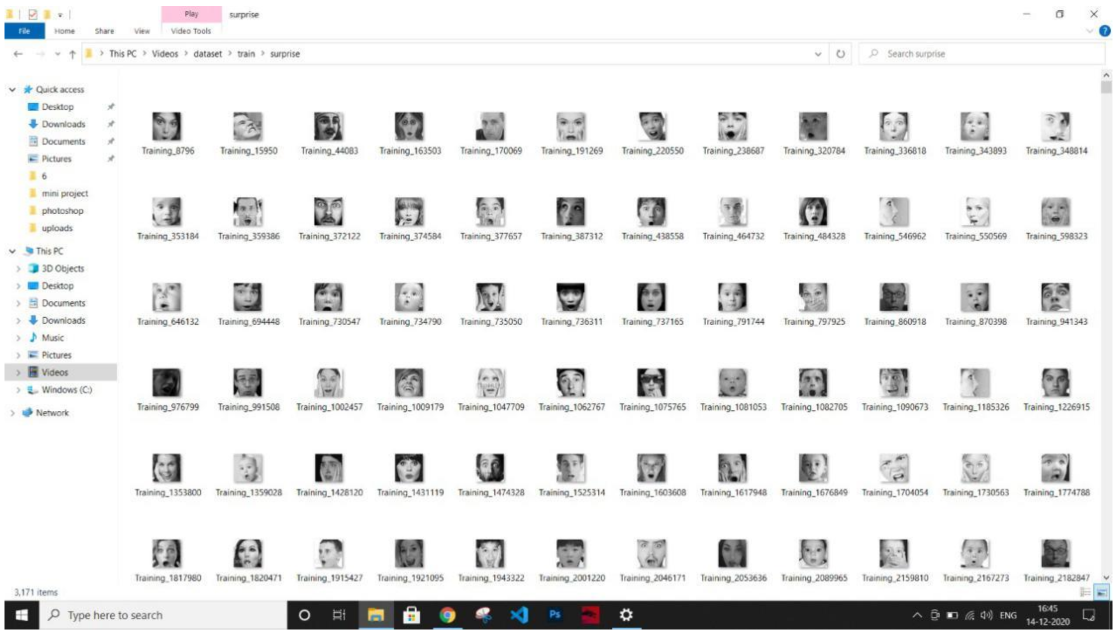


- **Expression Dataset**

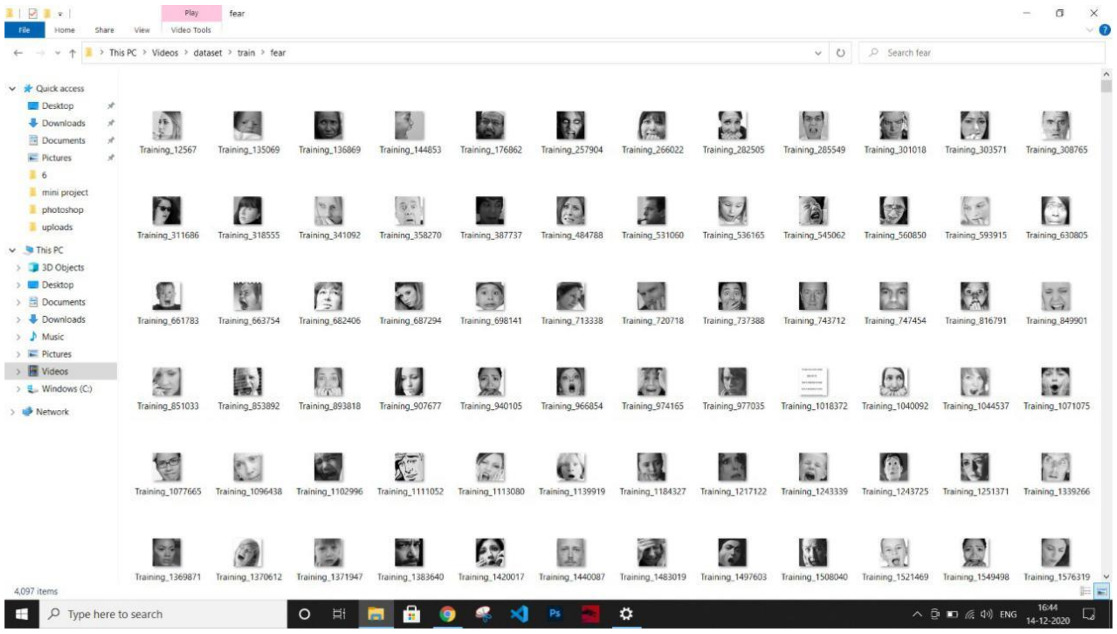
A folder of training and testing images containing the expressions that are evaluated for training sand testing by the model.Here is the portion of the training set of images used for training in the model for Gender detection, Expression detection, and Mask detection.



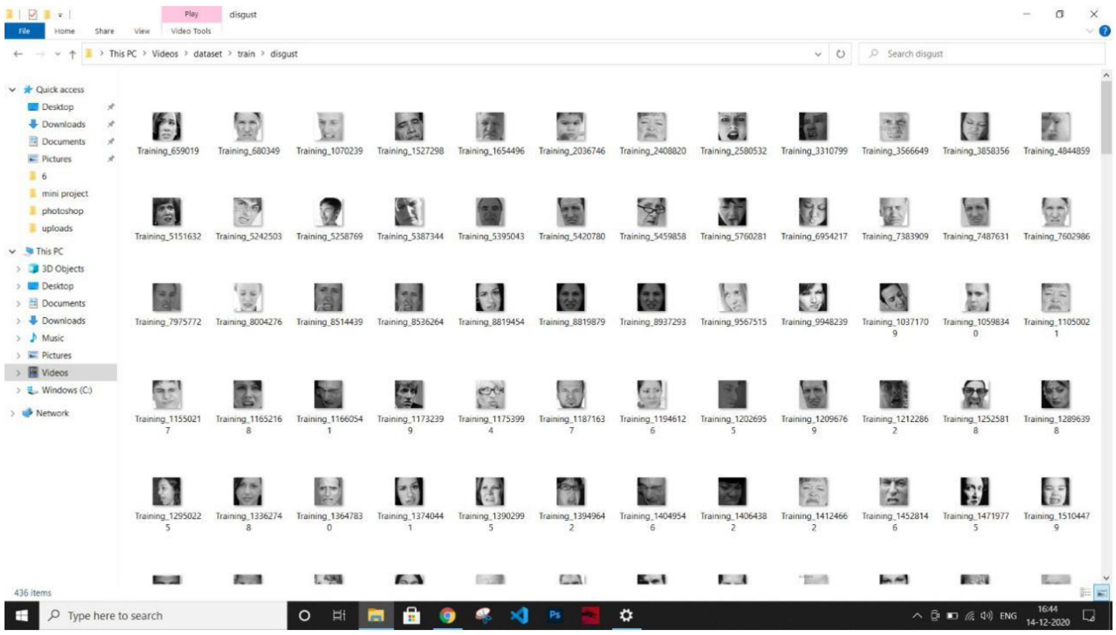
(a): Train/Angry



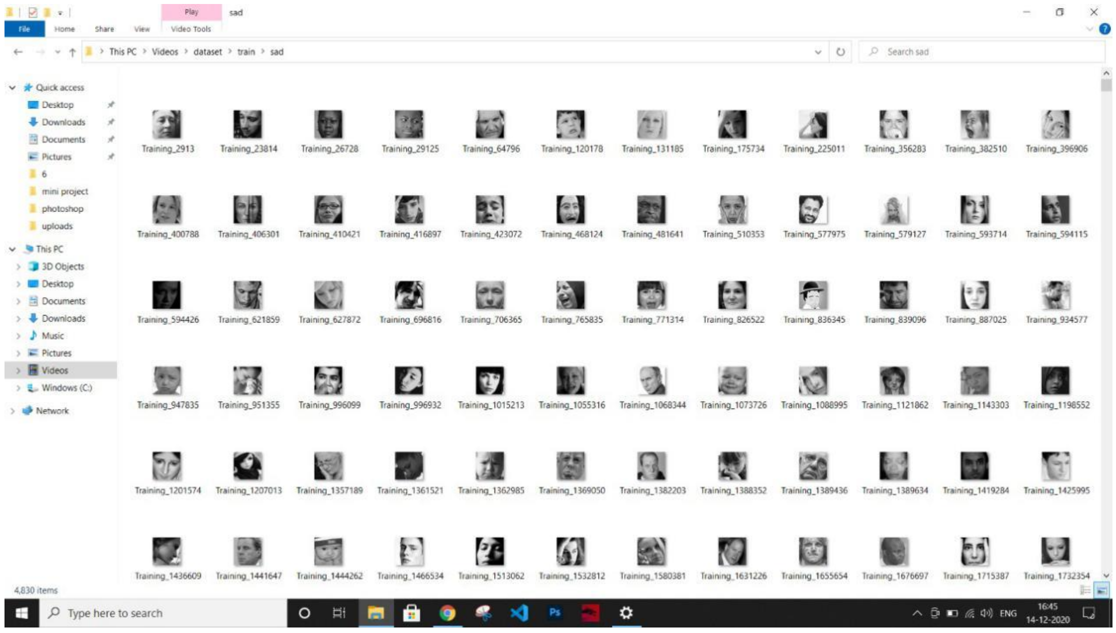
(b): Train/Surprise



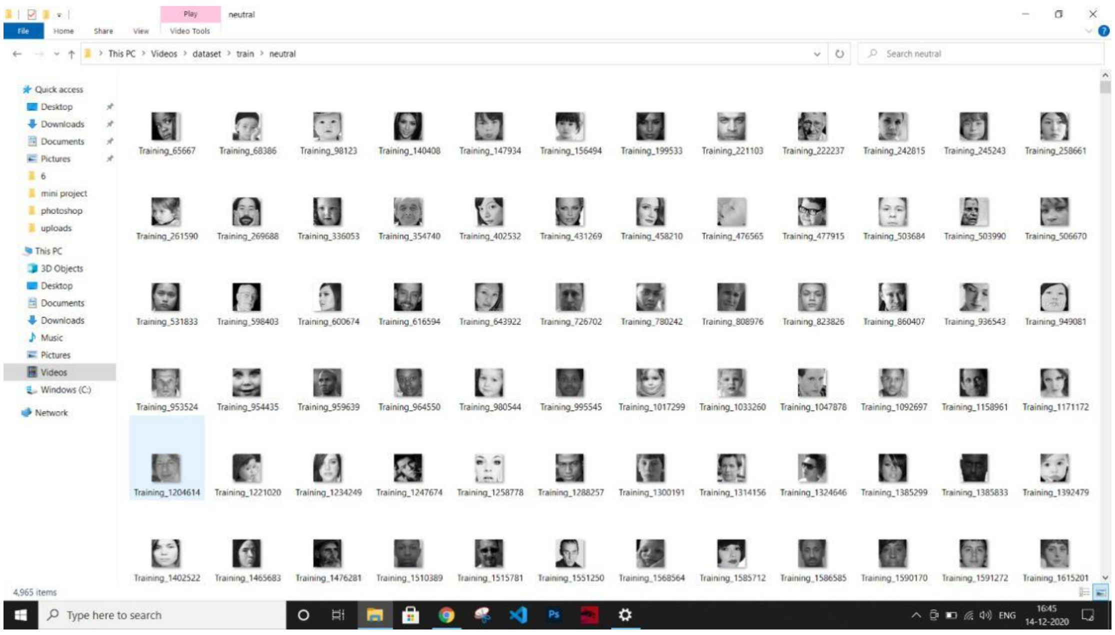
(c):Train/Fear



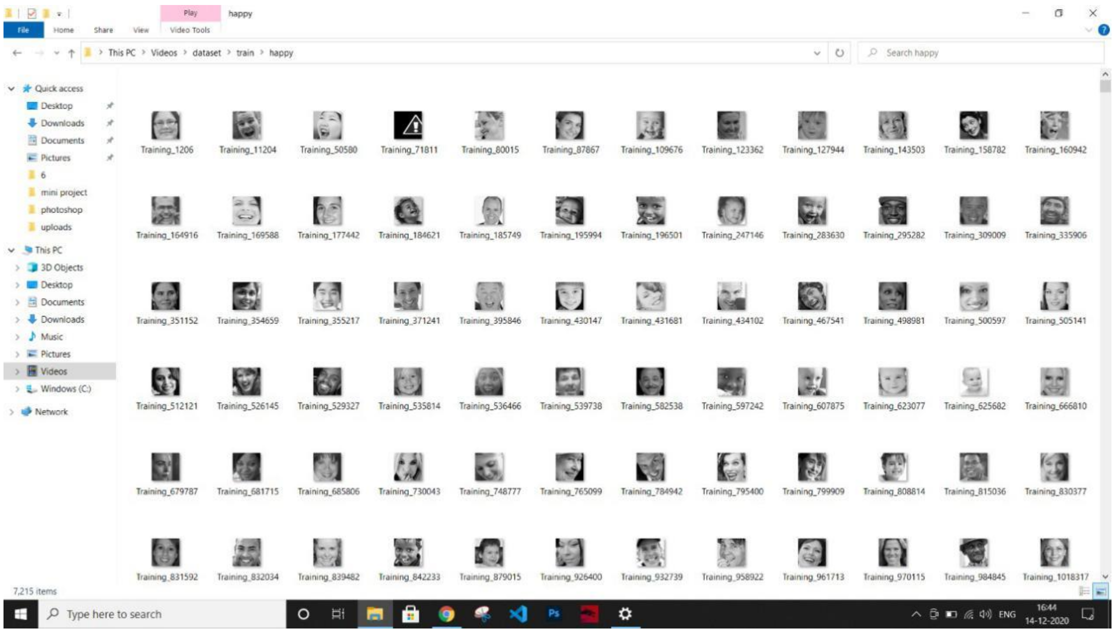
(d): Train/Disgust



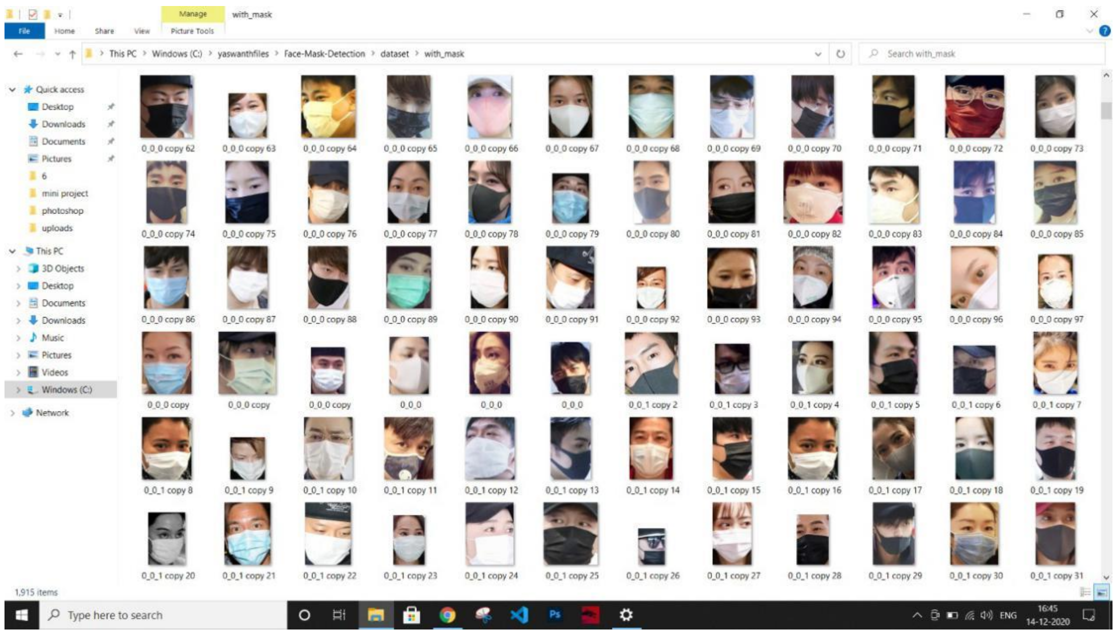
(e): Train/Sad



(f): Train/Neutral

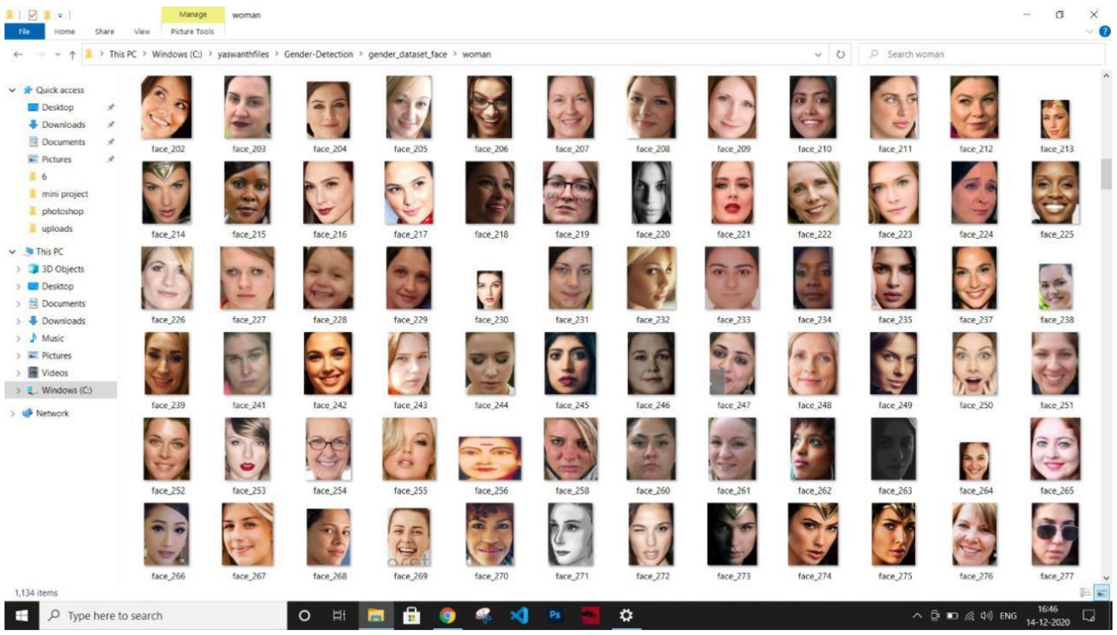


(g): Train/Happy

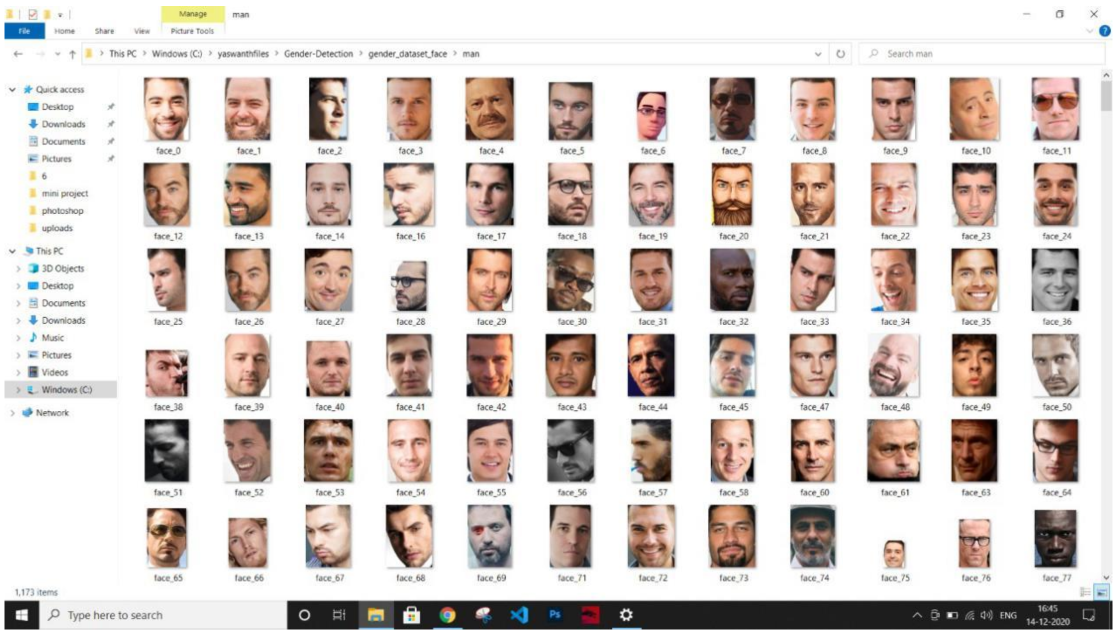


(h): Mask dataset containing images of persons wearing mask

- **Gender Dataset:**



(i): Gender_Dataset/Female


This PC > Windows (C:) > yaswanthfiles > Gender-Detection > gender_dataset_face > man
1,173 items

(j): Gender_Dataset/Male

Models that are created after training phase:

1. emotion.h5
2. genderdetect.h5
3. maskdetect.h5

7. REQUIREMENTS AND SPECIFICATIONS

Requirements play a vital role in the software development life cycle (SDLC).as it describes the complete requirements of the system, it is meant for use by the developers and will be the basic during testing phase. Any changes made to the future will have to go through formal change

Software Requirements Specifications:

1. FUNCTIONAL REQUIREMENTS

The functional requirements can be further Categorized as follows:

• What input the system should accept and under what conditions. Thisincludes data and command both from the user and the other system.

• What computations the system should produce.

A functional requirements defines a function of a system or it component. A function described as a set of inputs, the behavior, and outputs.

2. NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe user-visible aspects of the system that are not directly related to functionally of the system. These requirements define what qualities are exhibited by the system.

The following are the non function requirements of the system:

1. Performance (arrives at solution faster than hard computing approach)
2. Reliability
3. Use case
4. Maintainability
5. Interoperability

3. HARDWARE REQUIREMENTS

1. Processor : Intel Processor machine
2. Speed 1 : 1Ghz
3. RAM : 8GB
4. Hard Disk : 110GB
5. Input Devices : Keyboard, Mouse, web camera
6. Output Devices :Monitor

SOFTWARE REQUIREMENTS

1. Operating system: Windows 7,8,10
2. Programming language: python
3. IDE's -visual studio code, atom, spider, jupyternotebook
4. Open-cv framework
5. Window operating system
6. Tensor flow package
7. Flask package
8. Sql-lite package

8. PROGRAMMING LANGUAGE

Python Programming Language is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C++. The language founded in the year 1991 by the developer Guido Van Rossum has the programming easy and fun to do. The Python is widely used in bigger organizations because of its multiple programming paradigms. They usually involve imperative and object-oriented functional programming. It has a comprehensive and large standard library that has automatic memory management and dynamic features.

Why Python?

Python has topped the charts in recent years over other programming languages like C, C++, and Java and is widely used by the programmers. The language has undergone a drastic change since its release 25 years ago as many add-on features are introduced. The Python 1.0 had the module system of Modula-3 and interacted with Amoeba Operating System with varied functioning tools. Python 2.0 introduced in the year 2000 had features of the garbage collector and Unicode Support. Python 3.0 introduced in the year 2008 had a constructive design that avoids duplicate modules and constructs. With the added features, now the companies are using Python 3.5.

The software development companies prefer Python language because of its versatile features and fewer programming codes. Nearly 14% of the programmers use it on the operating systems like UNIX, Linux, Windows and Mac OS. The programmers of big companies use Python as it has created a mark for itself in the software development with characteristic features like

•Interactive
•Interpreted
•Modular
•Dynamic
•Object-oriented
•Portable
•High level

•Extensible in C++ & C

Advantages of Python

The Python language has diversified application in the software development companies such as in gaming, web frameworks and applications, language development, prototyping, graphic design applications, etc. This provides the language a higher plethora over other programming languages used in the industry. Some of its advantages are:

1. Extensive Support Libraries

It provides large standard libraries that include areas like string operations, Internet, web service tools, operating system interfaces, and protocols. Most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python.

2. Integration Feature

Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Jython. Python also processes XML and other mark-up languages as it can run on all modern operating systems through same byte code.

Limitations of Python

Python has varied advantageous features, and programmers prefer this language to other programming languages because it is easy to learn and code too. However, this language has still not made its place in some computing arenas that includes Enterprise Development Shops. Therefore, this language may not solve some of the enterprise solutions, and limitations include-

1. Difficulty in Using Other Languages

The Python lovers become so accustomed to its features and its extensive libraries, so they face problem in learning or working on other programming languages. Python experts may see the declaring of cast "values" or variable "types", syntactic requirements of adding curly braces or semi-colons as an onerous task.

2. Weak in Mobile Computing

Python has made its presence on many desktop and server platforms, but it is seen as a weak language for mobile computing. This is the reason very few mobile applications are built in it like Carbonnelle.

3. Gets Slow in Speed

Python executes with the help of an interpreter instead of the compiler, which causes it to slow down because compilation and execution help it to work normally. On the other hand, it can be seen that it is fast for many web applications too.

OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

INSTALLATION OF OPENCV:-

To install open cv, we need to have Python and Pip installed.

- Open CV can be directly downloaded and installed with the use of pip.

To install opencv, just go to the command prompt and type the command:

Pip install opencv-python

This will automatically download and will be installed using the pip command in the command prompt.

BASIC OPERATIONS WITH OPENCV:-

i. Loading an image

To load an image, first we need to import opencv and then load an image.
We use the syntax:
cv2.imread (image)

ii. Displaying an image

To display an image, we use the syntax:
cv2.imshow (window name, image)

iii. Writing an image

To write an image, we use the syntax:
cv2.imwrite (filename, image)

iv. Capturing a video

To capture a video, we use the syntax:
cv2.VideoCapture (0) Here 0 indicates Web camera.

TENSORFLOW

TensorFlow is an open source customizable software library for performing numerical and graphical computations using data flow graphs. A flexible, scalable, and portable system used for creating large-scale neural networks with multiple layers. The base language for TensorFlow is Python or C++.TensorFlow provides fantastic architectural support that make it easy to deploy complex numerical computations across diverse platforms ranging from PC's to mobiles, edge devices, and also cluster of servers. TensorFlow has been designed for use both in research and development and in production systems. TensorFlow might be overkill for simpler tasks but a strong bet for complex deep learning tasks.

TENSORFLOW ARCHITECTURE

Tensorflow architecture works in three parts:

- Preprocessing the data
- Build the model
- Train and estimate the model

It is called Tensorflow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

FLASK

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, and a wiki or go as big as a web-based calendar application or a commercial website.

Flask is part of the categories of the micro-framework. Micro-framework is normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plug-in.

A minimal Flask application looks something like this:

```
fromflaskimportFlask
                              app=Flask(__name__)

                              @app.route('/')
defhello_world():
return'Hello World!'

                              if__name__=='__main__':
app.run()
```

Training the expression model with dataset:

1. Importing the packages

```
import numpy as np
import cv2

from tensorflow.keras.models import Sequential #for initializing
from tensorflow.keras.layers import Dense  #adding layers
from tensorflow.keras.layers import Conv2D  #adding convolution layer
from tensorflow.keras.layers import MaxPooling2D  #max pooling
from tensorflow.keras.layers import Flatten,Dropout,Flatten
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
```

2. Loading the classifying the Dataset

```
train_dir = 'data/train'
val_dir = 'data/test'
train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(48,48),
        batch_size=64,
        color_mode="grayscale",
        class_mode='categorical')
```

```
Found 28709 images belonging to 7 classes.
```

3. Creating the Model

```
emotion_model = Sequential()
```

```
emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
emotion_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))
```

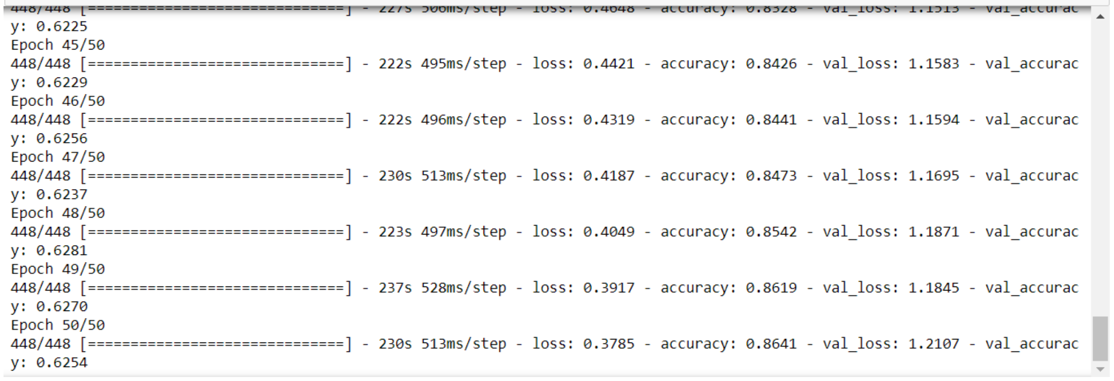
```
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))
```

```
emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))
```

```
cv2.ocl.setUseOpenCL(False)
emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral", 5: "Sad", 6: "Surprised"}
```

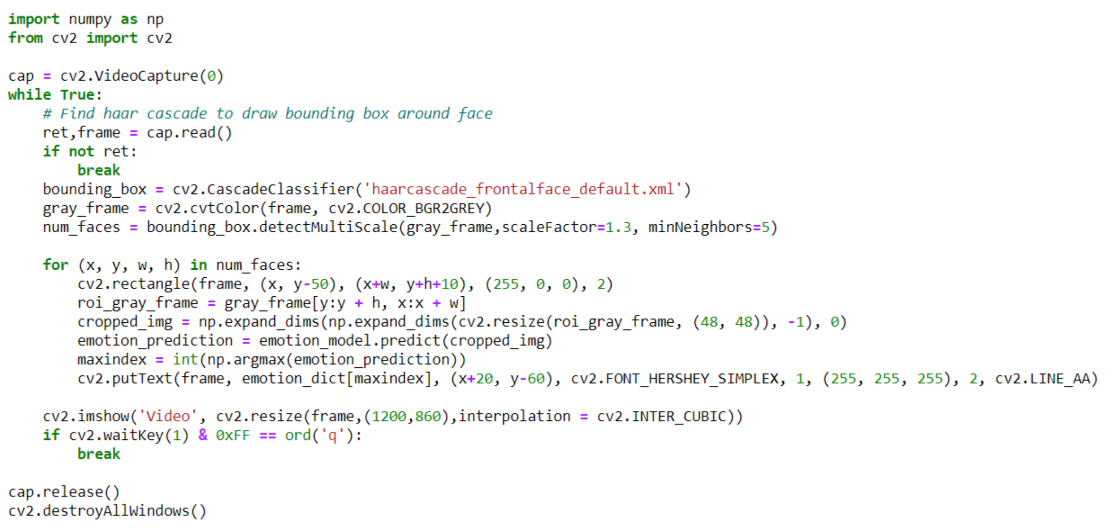
4. Training the model using the dataset

```
emotion_model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.0001, decay=1e-6),metrics=['accuracy'])
emotion_model_info = emotion_model.fit_generator(
        train_generator,
        steps_per_epoch=28709 // 64,
        epochs=50,
        validation_data=validation_generator,
        validation_steps=7178 // 64)
emotion_model.save_weights('emotion_model.h5')
```



```
448/448 [==============================] - 227s 506ms/step - loss: 0.4648 - accuracy: 0.8528 - val_loss: 1.1513 - val_accurac
y: 0.6225
Epoch 45/50
448/448 [==============================] - 222s 495ms/step - loss: 0.4421 - accuracy: 0.8426 - val_loss: 1.1583 - val_accurac
y: 0.6229
Epoch 46/50
448/448 [==============================] - 222s 496ms/step - loss: 0.4319 - accuracy: 0.8441 - val_loss: 1.1594 - val_accurac
y: 0.6256
Epoch 47/50
448/448 [==============================] - 230s 513ms/step - loss: 0.4187 - accuracy: 0.8473 - val_loss: 1.1695 - val_accurac
y: 0.6237
Epoch 48/50
448/448 [==============================] - 223s 497ms/step - loss: 0.4049 - accuracy: 0.8542 - val_loss: 1.1871 - val_accurac
y: 0.6281
Epoch 49/50
448/448 [==============================] - 237s 528ms/step - loss: 0.3917 - accuracy: 0.8619 - val_loss: 1.1845 - val_accurac
y: 0.6270
Epoch 50/50
448/448 [==============================] - 230s 513ms/step - loss: 0.3785 - accuracy: 0.8641 - val_loss: 1.2107 - val_accurac
y: 0.6254
```

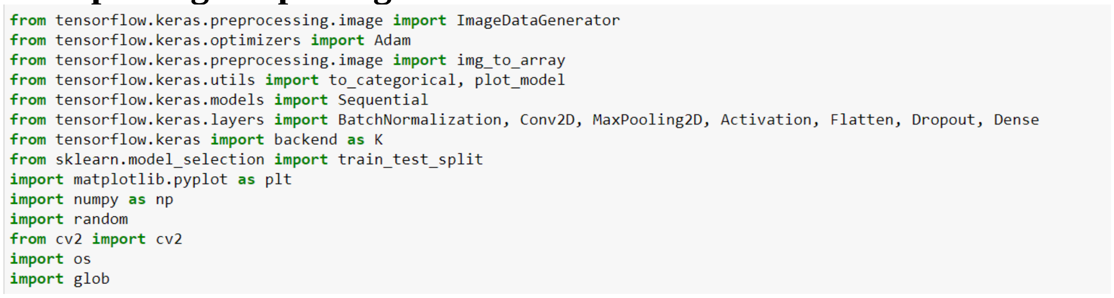
5. Predicting the expression using the model



```python
import numpy as np
from cv2 import cv2

cap = cv2.VideoCapture(0)
while True:
    # Find haar cascade to draw bounding box around face
    ret,frame = cap.read()
    if not ret:
        break
    bounding_box = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GREY)
    num_faces = bounding_box.detectMultiScale(gray_frame,scaleFactor=1.3, minNeighbors=5)

    for (x, y, w, h) in num_faces:
        cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (255, 0, 0), 2)
        roi_gray_frame = gray_frame[y:y + h, x:x + w]
        cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray_frame, (48, 48)), -1), 0)
        emotion_prediction = emotion_model.predict(cropped_img)
        maxindex = int(np.argmax(emotion_prediction))
        cv2.putText(frame, emotion_dict[maxindex], (x+20, y-60), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)

    cv2.imshow('Video', cv2.resize(frame,(1200,860),interpolation = cv2.INTER_CUBIC))
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

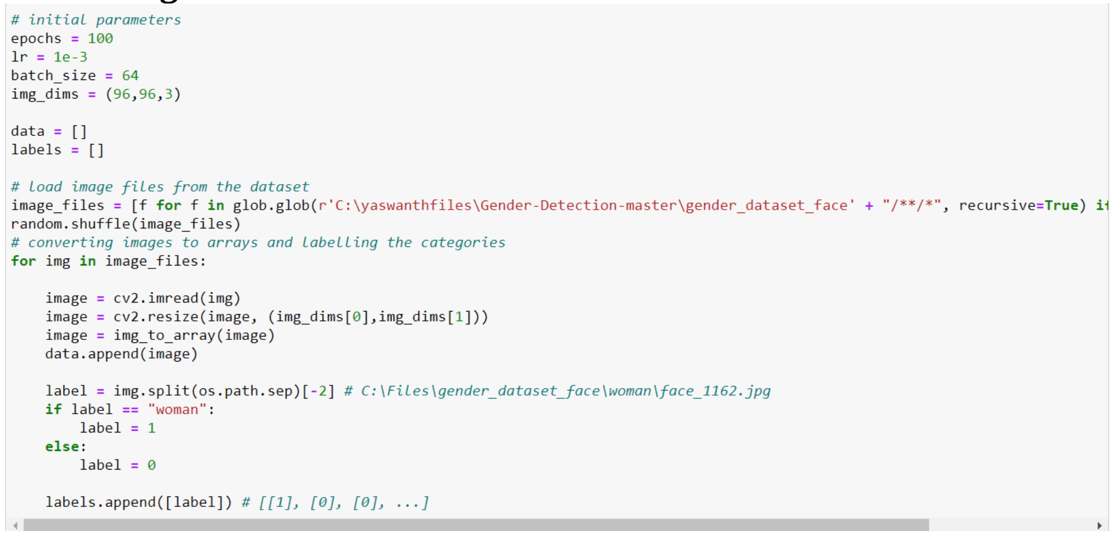
• Gender Detection Model

1. Importing the packages



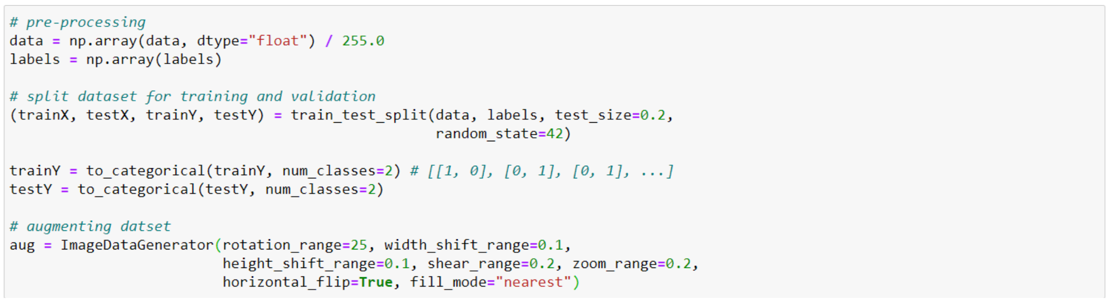
```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.utils import to_categorical, plot_model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import BatchNormalization, Conv2D, MaxPooling2D, Activation, Flatten, Dropout, Dense
from tensorflow.keras import backend as K
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np
import random
from cv2 import cv2
import os
import glob
```

2. Loading the Dataset



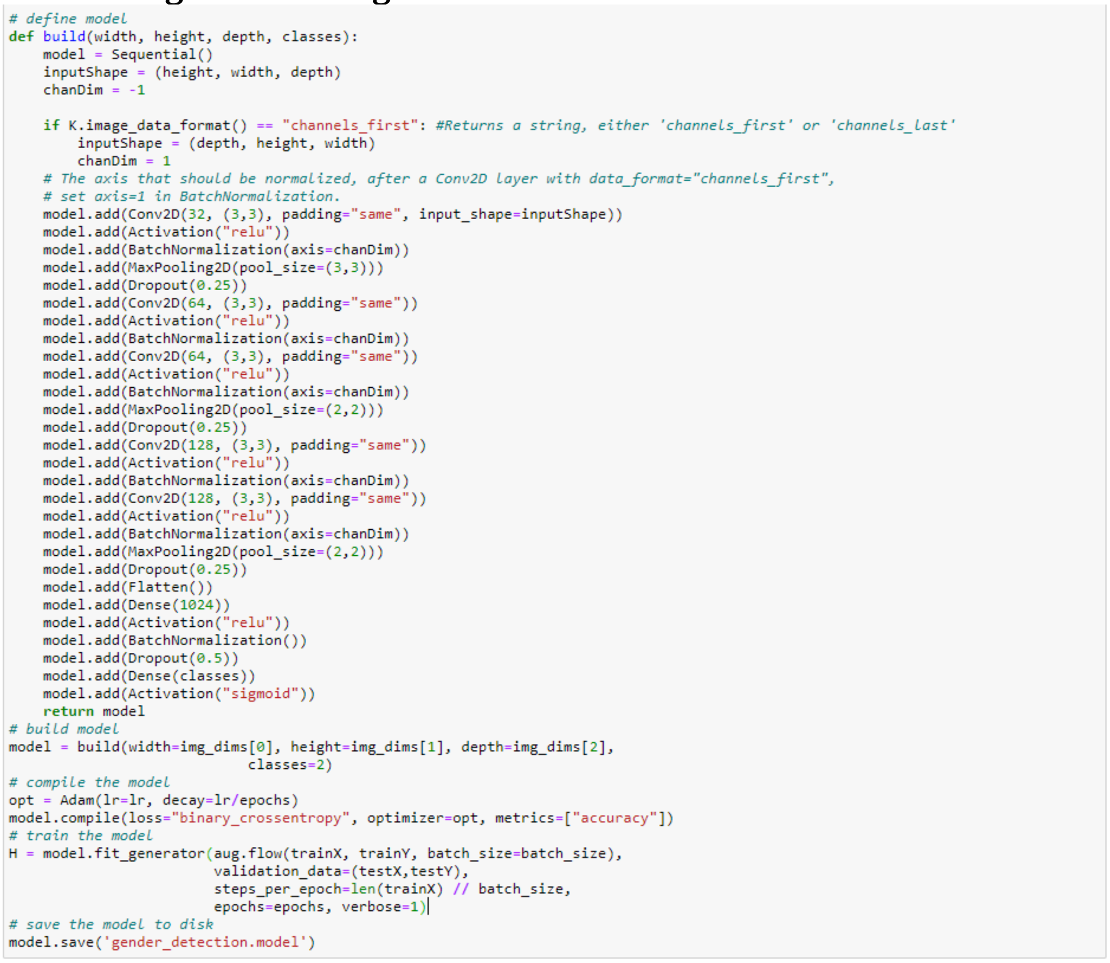
```python
# initial parameters
epochs = 100
lr = 1e-3
batch_size = 64
img_dims = (96,96,3)

data = []
labels = []

# load image files from the dataset
image_files = [f for f in glob.glob(r'C:\yaswanthfiles\Gender-Detection-master\gender_dataset_face' + "/**/*", recursive=True) i
random.shuffle(image_files)
# converting images to arrays and labelling the categories
for img in image_files:

    image = cv2.imread(img)
    image = cv2.resize(image, (img_dims[0],img_dims[1]))
    image = img_to_array(image)
    data.append(image)

    label = img.split(os.path.sep)[-2] # C:\Files\gender_dataset_face\woman\face_1162.jpg
    if label == "woman":
        label = 1
    else:
        label = 0

    labels.append([label]) # [[1], [0], [0], ...]
```

3. Processing and augmenting the dataset



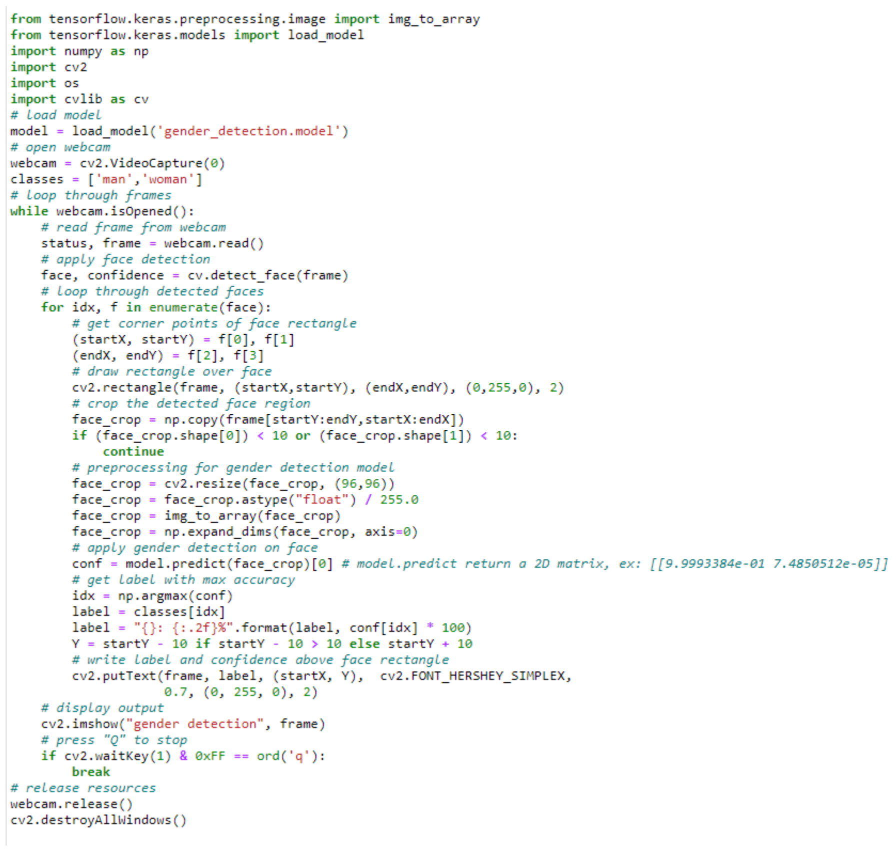
```python
# pre-processing
data = np.array(data, dtype="float") / 255.0
labels = np.array(labels)

# split dataset for training and validation
(trainX, testX, trainY, testY) = train_test_split(data, labels, test_size=0.2,
                                                  random_state=42)

trainY = to_categorical(trainY, num_classes=2) # [[1, 0], [0, 1], [0, 1], ...]
testY = to_categorical(testY, num_classes=2)

# augmenting datset
aug = ImageDataGenerator(rotation_range=25, width_shift_range=0.1,
                         height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
                         horizontal_flip=True, fill_mode="nearest")
```

4. Creating and training the model



```python
# define model
def build(width, height, depth, classes):
    model = Sequential()
    inputShape = (height, width, depth)
    chanDim = -1

    if K.image_data_format() == "channels_first": #Returns a string, either 'channels_first' or 'channels_last'
        inputShape = (depth, height, width)
        chanDim = 1
    # The axis that should be normalized, after a Conv2D layer with data_format="channels_first",
    # set axis=1 in BatchNormalization.
    model.add(Conv2D(32, (3,3), padding="same", input_shape=inputShape))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(3,3)))
    model.add(Dropout(0.25))
    model.add(Conv2D(64, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(Conv2D(64, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25))
    model.add(Conv2D(128, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(Conv2D(128, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25))
    model.add(Flatten())
    model.add(Dense(1024))
    model.add(Activation("relu"))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(classes))
    model.add(Activation("sigmoid"))
    return model
# build model
model = build(width=img_dims[0], height=img_dims[1], depth=img_dims[2],
                            classes=2)
# compile the model
opt = Adam(lr=lr, decay=lr/epochs)
model.compile(loss="binary_crossentropy", optimizer=opt, metrics=["accuracy"])
# train the model
H = model.fit_generator(aug.flow(trainX, trainY, batch_size=batch_size),
                        validation_data=(testX,testY),
                        steps_per_epoch=len(trainX) // batch_size,
                        epochs=epochs, verbose=1)
# save the model to disk
model.save('gender_detection.model')
```

5. Predicting the Gender using model



```python
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
import numpy as np
import cv2
import os
import cvlib as cv
# load model
model = load_model('gender_detection.model')
# open webcam
webcam = cv2.VideoCapture(0)
classes = ['man','woman']
# loop through frames
while webcam.isOpened():
    # read frame from webcam
    status, frame = webcam.read()
    # apply face detection
    face, confidence = cv.detect_face(frame)
    # loop through detected faces
    for idx, f in enumerate(face):
        # get corner points of face rectangle
        (startX, startY) = f[0], f[1]
        (endX, endY) = f[2], f[3]
        # draw rectangle over face
        cv2.rectangle(frame, (startX,startY), (endX,endY), (0,255,0), 2)
        # crop the detected face region
        face_crop = np.copy(frame[startY:endY,startX:endX])
        if (face_crop.shape[0]) < 10 or (face_crop.shape[1]) < 10:
            continue
        # preprocessing for gender detection model
        face_crop = cv2.resize(face_crop, (96,96))
        face_crop = face_crop.astype("float") / 255.0
        face_crop = img_to_array(face_crop)
        face_crop = np.expand_dims(face_crop, axis=0)
        # apply gender detection on face
        conf = model.predict(face_crop)[0] # model.predict return a 2D matrix, ex: [[9.9993384e-01 7.4850512e-05]]
        # get label with max accuracy
        idx = np.argmax(conf)
        label = classes[idx]
        label = "{}: {:.2f}%".format(label, conf[idx] * 100)
        Y = startY - 10 if startY - 10 > 10 else startY + 10
        # write label and confidence above face rectangle
        cv2.putText(frame, label, (startX, Y),  cv2.FONT_HERSHEY_SIMPLEX,
                    0.7, (0, 255, 0), 2)
    # display output
    cv2.imshow("gender detection", frame)
    # press "Q" to stop
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
# release resources
webcam.release()
cv2.destroyAllWindows()
```

9. FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and DesignProcess. The study begins by classifying the problem definition. Feasibility is to determine ifit's worth doing. Once an acceptance problem definition has been generated, the analystdevelops a logical model of the system. A search for alternatives is analyzed carefully. Thereare 3 parts in feasibility study.

1. Operational Feasibility

Question that going to be asked are:

- Will the system be used if it developed and implemented?
- If there was sufficient support for the project from the management and from the users.
- Have the users been involved in planning and development of the Project.
- Will the system produce poorer result in any respect or area?

This system can be implemented in the organization because there is adequate support frommanagement and users. Being developed in Python so that the necessary operations are carriedout automatically.

2. Technical feasibility

- Does the necessary technology exist to do what is been suggested?
- Does the proposed equipment have the technical capacity for using the new system?
- Are there technical guarantees of accuracy, reliability and data security?
- The project is developed on Pentium IV with 256 MB RAM.
- The environment required in the development of system is any windows platform
- The observer pattern along with factory pattern will update the results eventually
- The language used in the development is PYTHON 3.6 & Windows Environment.

3. Financial and Economic Feasibility

The system developed and installed will be good benefit to the organization. The system will be developed and operated in the existing hardware and software infrastructure. So, there is noneed of additional hardware and software for the system.

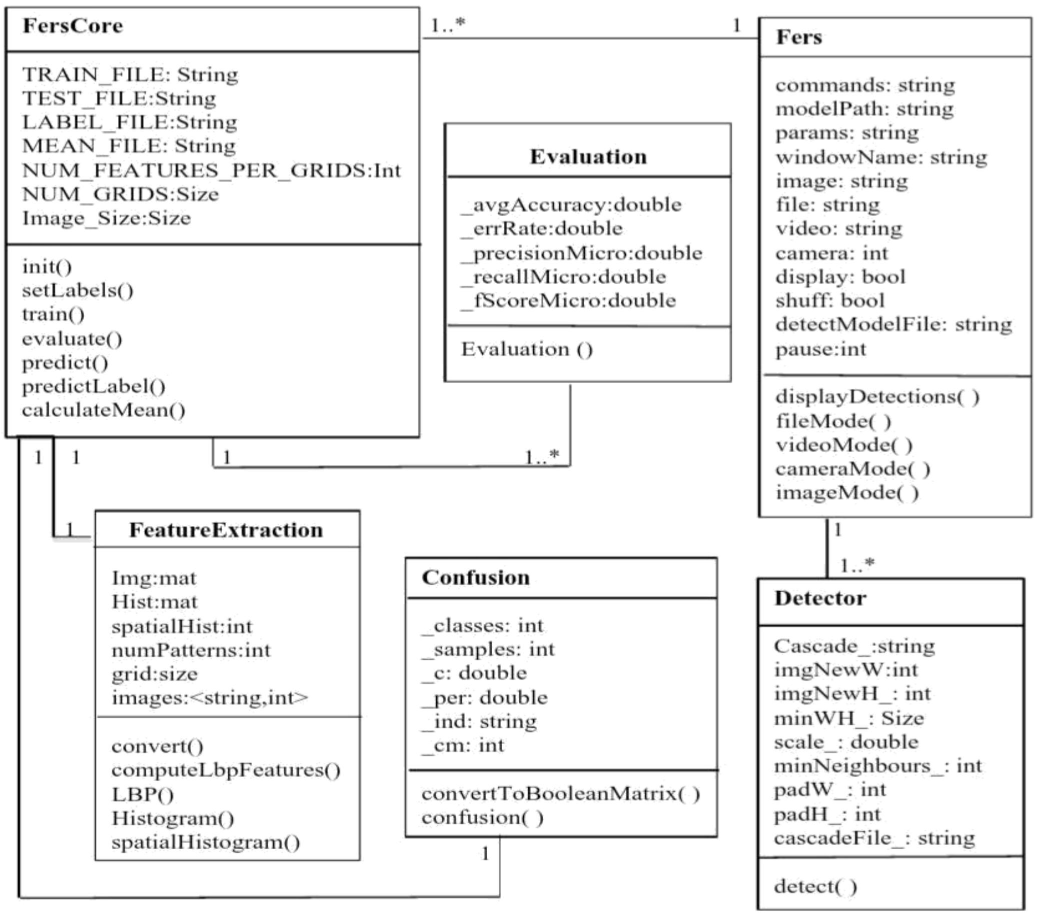
10. UML DESIGN

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

• UML stands for Unified Modeling Language.
• UML is different from the other common programming languages such as C++, Java, COBOL, etc...
• UML is a pictorial language used to make software blueprints.
• UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system.
• Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flows in a manufacturing unit, etc…

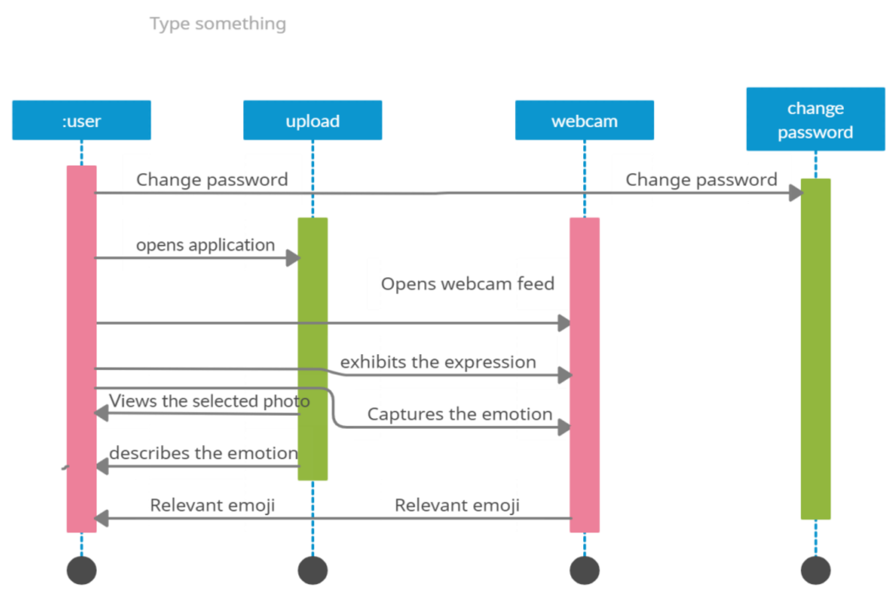
CLASS DIAGRAM

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. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, andconstraints. It is also known as a structural diagram.



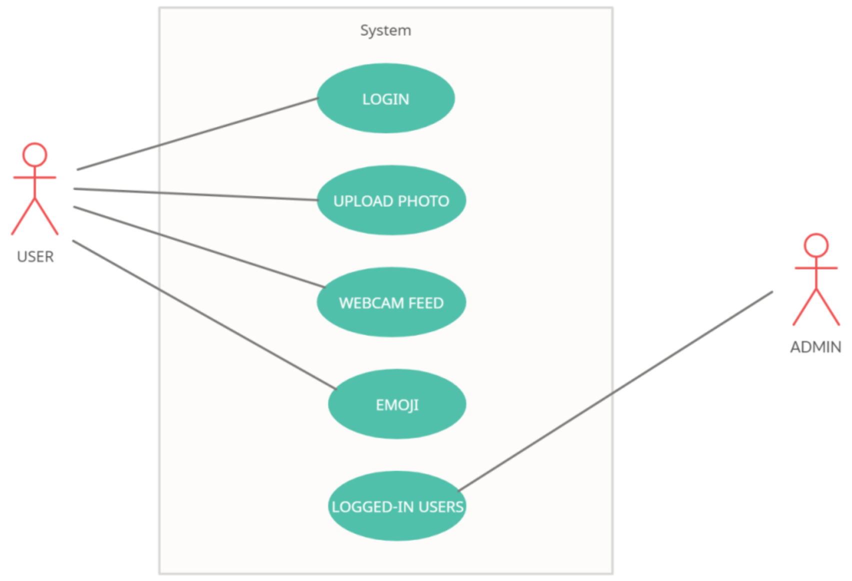
SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.



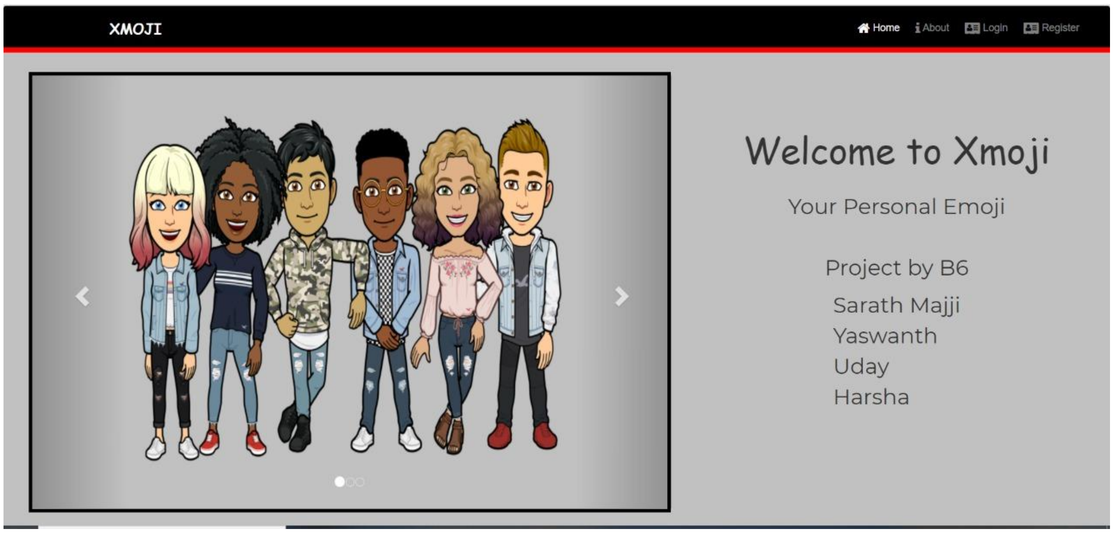
USE CASE DIAGRAM

A **UML** use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.



11.IMPLEMENTATION OF GUI USING FLASK

Home Page:



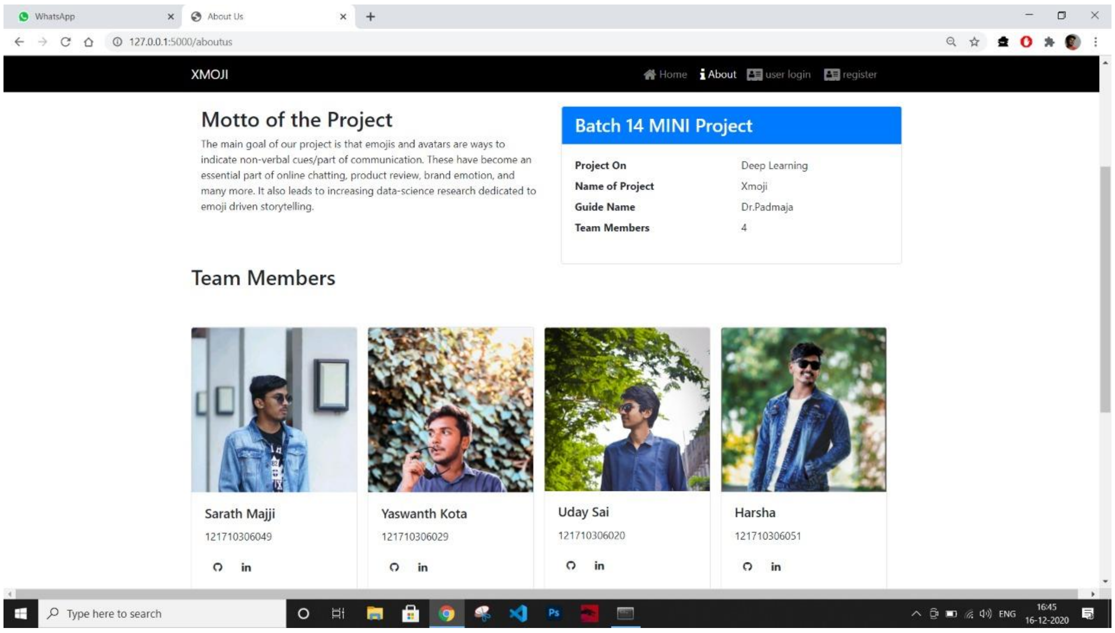
Welcome to Xmoji

Your Personal Emoji

Project by B6

Sarath Majji
Yaswanth
Uday
Harsha

About Us page:



Motto of the Project

The main goal of our project is that emojis and avatars are ways to indicate non-verbal cues/part of communication. These have become an essential part of online chatting, product review, brand emotion, and many more. It also leads to increasing data-science research dedicated to emoji driven storytelling.

Batch 14 MINI Project

Project On	Deep Learning
Name of Project	Xmoji
Guide Name	Dr.Padmaja
Team Members	4

Team Members

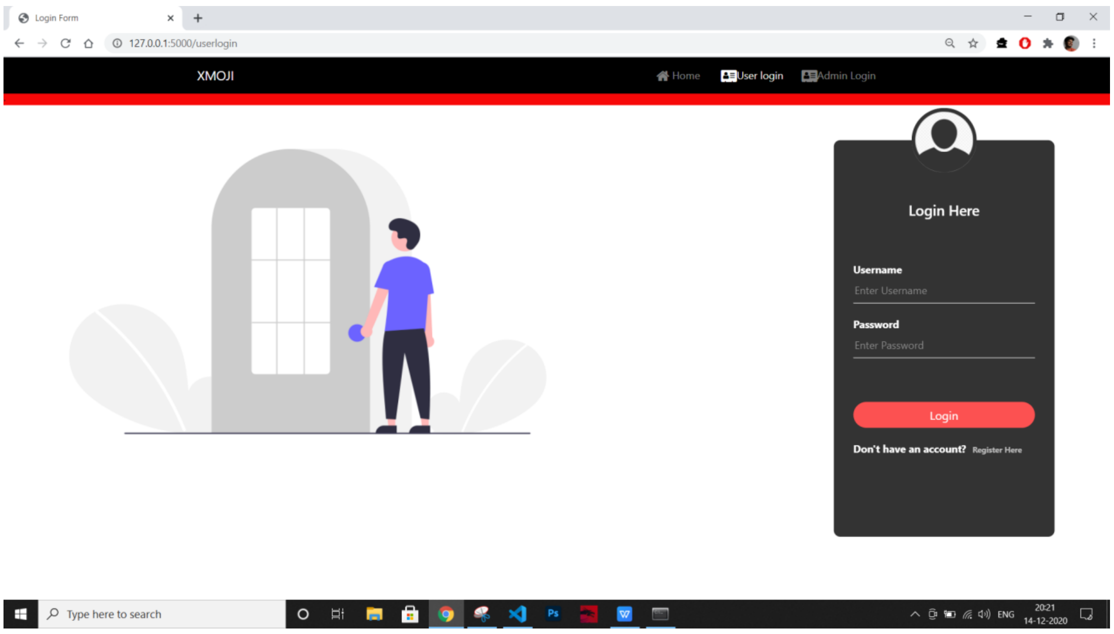
Sarath Majji
121710306049

Yaswanth Kota
121710306029

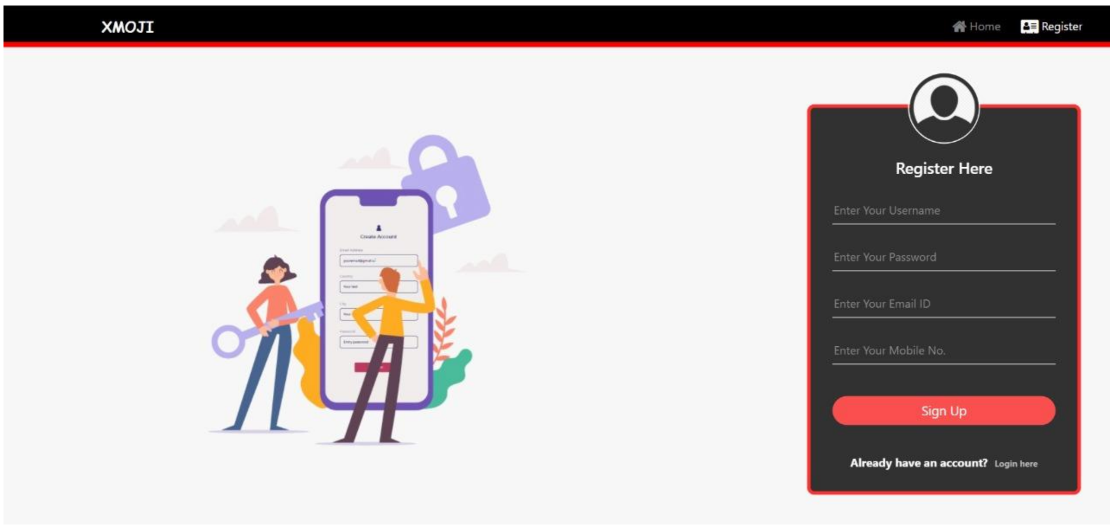
Uday Sai
121710306020

Harsha
121710306051

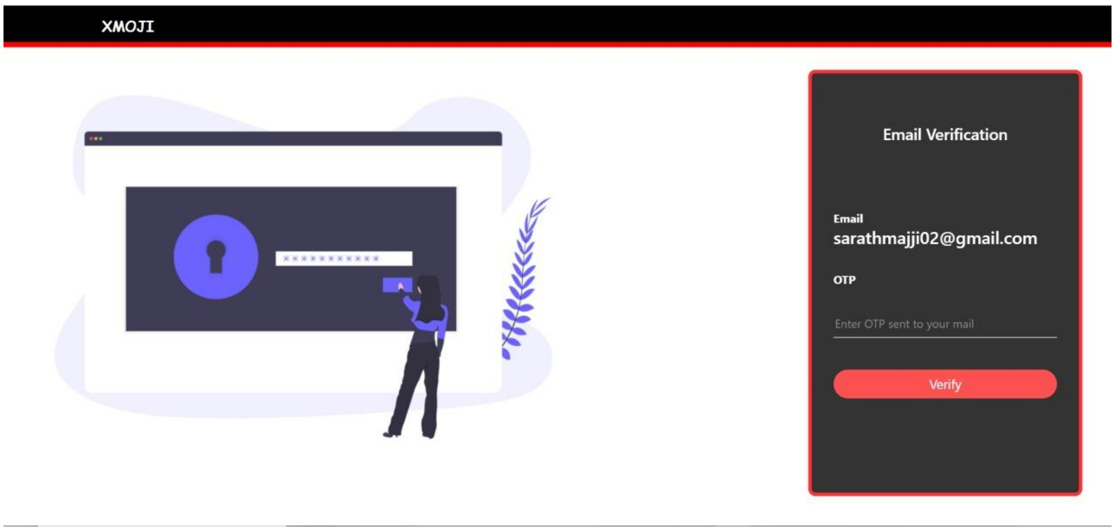
User Login Page:



User Registration Page:

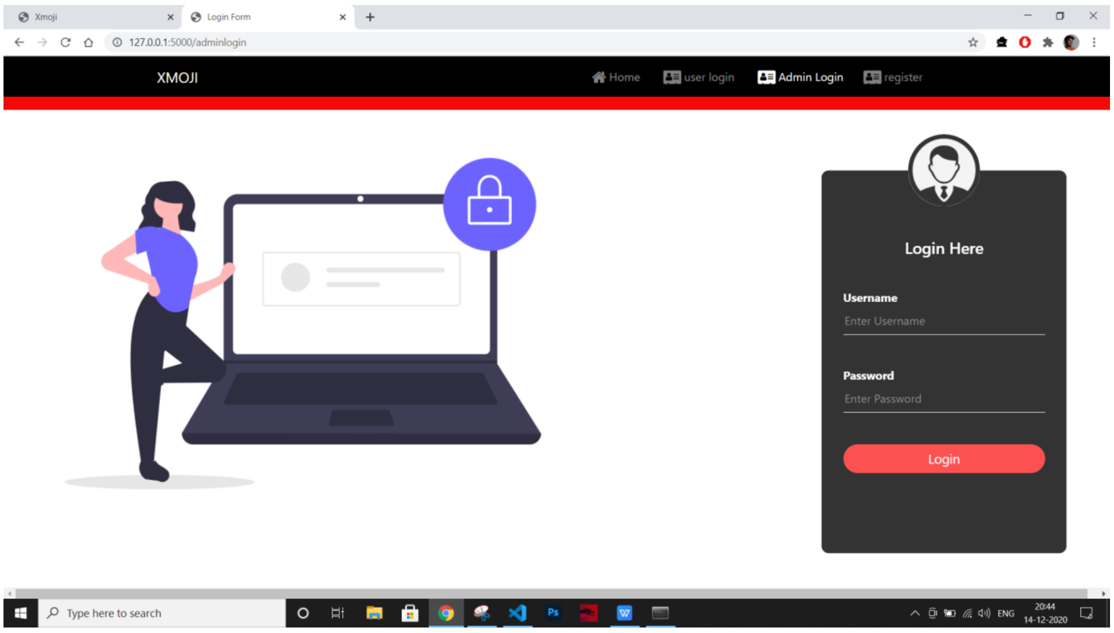


Registered Users Mail Verification:-

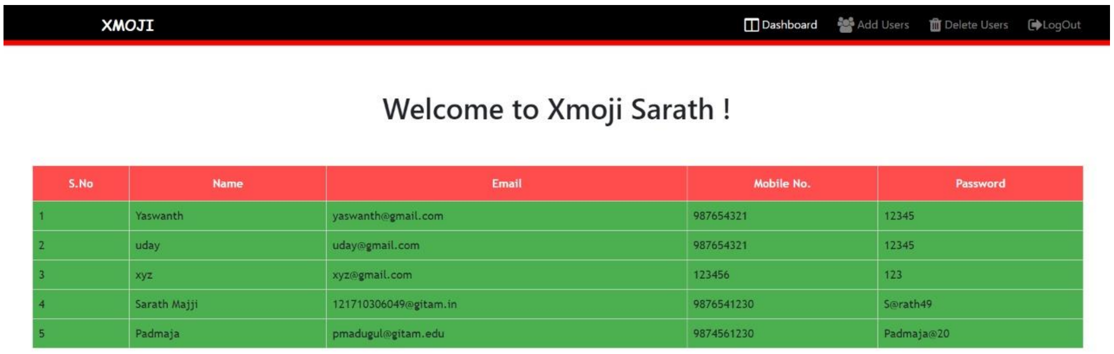


Email
sarathmajji02@gmail.com
OTP
Enter OTP sent to your mail

Admin Login Page:



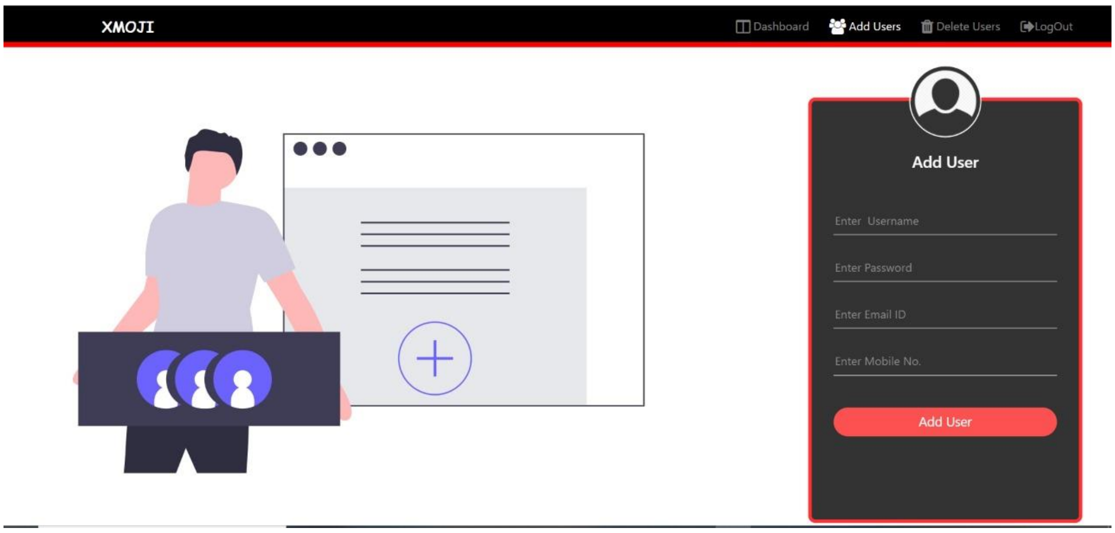
Registered users view to Admin:



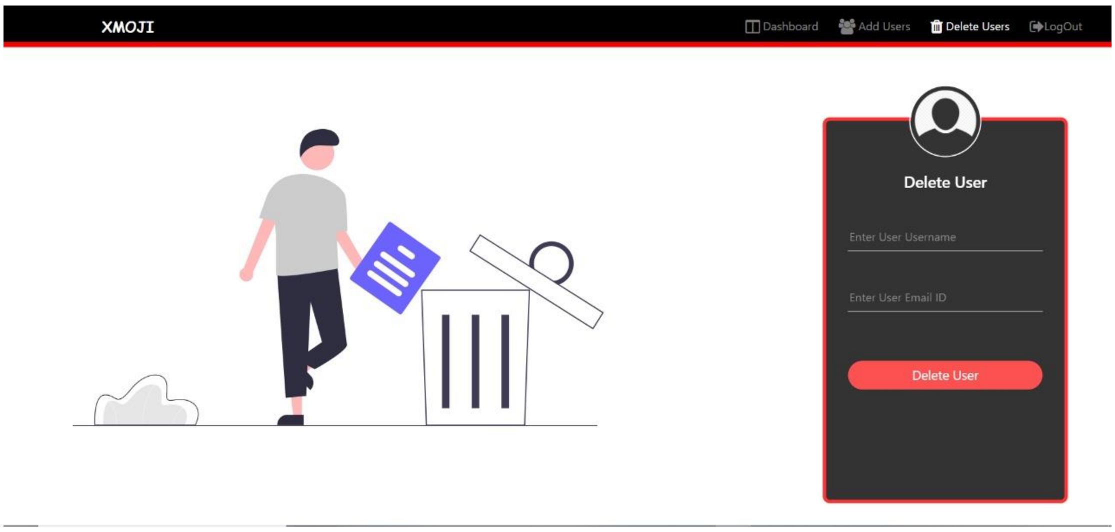
Welcome to Xmoji Sarath !

S.No	Name	Email	Mobile No.	Password
1	Yaswanth	yaswanth@gmail.com	987654321	12345
2	uday	uday@gmail.com	987654321	12345
3	xyz	xyz@gmail.com	123456	123
4	Sarath Majji	121710306049@gitam.in	9876541230	S@rath49
5	Padmaja	pmadugul@gitam.edu	9874561230	Padmaja@20

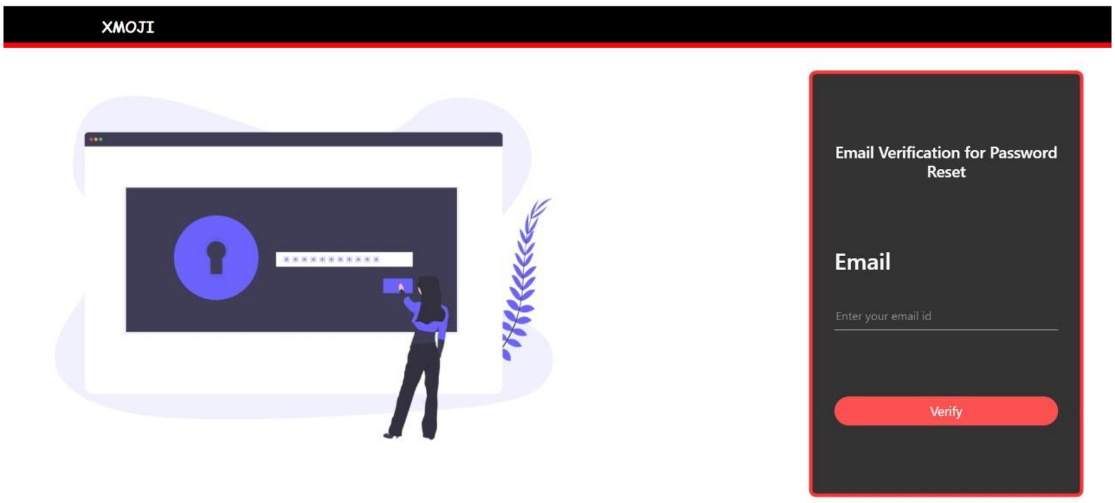
Add Users :-



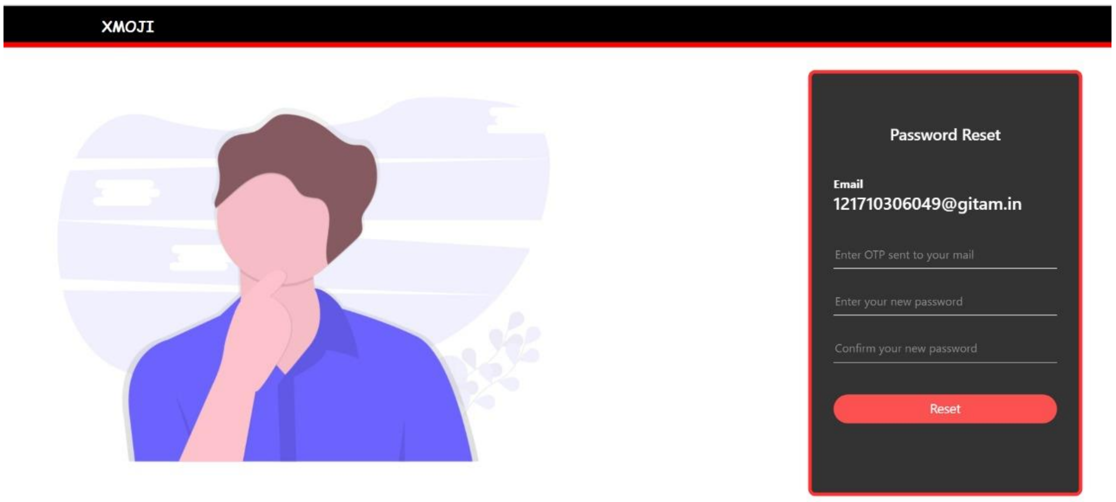
Delete Users:-



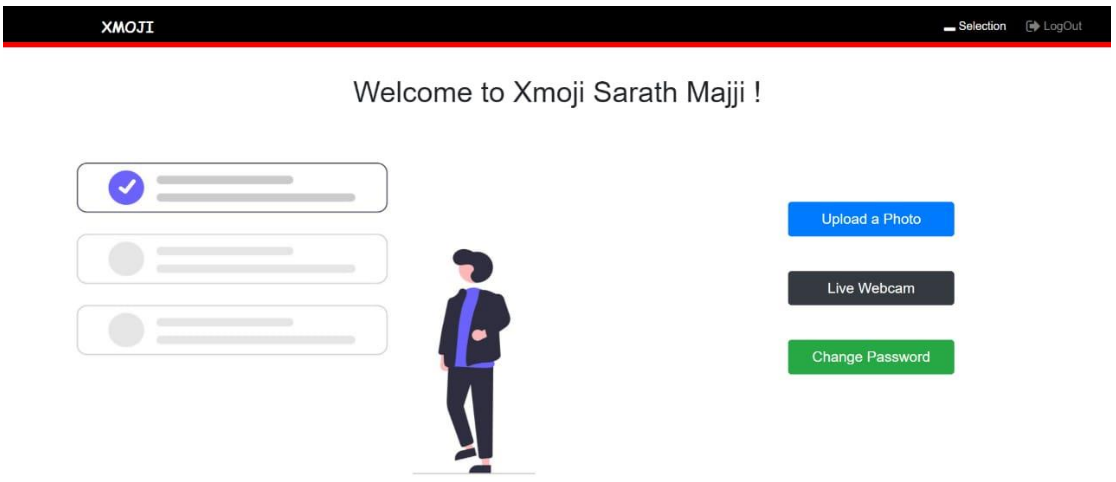
Password reset:



OTP for Password Reset:-



Selection Page:



Changing password:

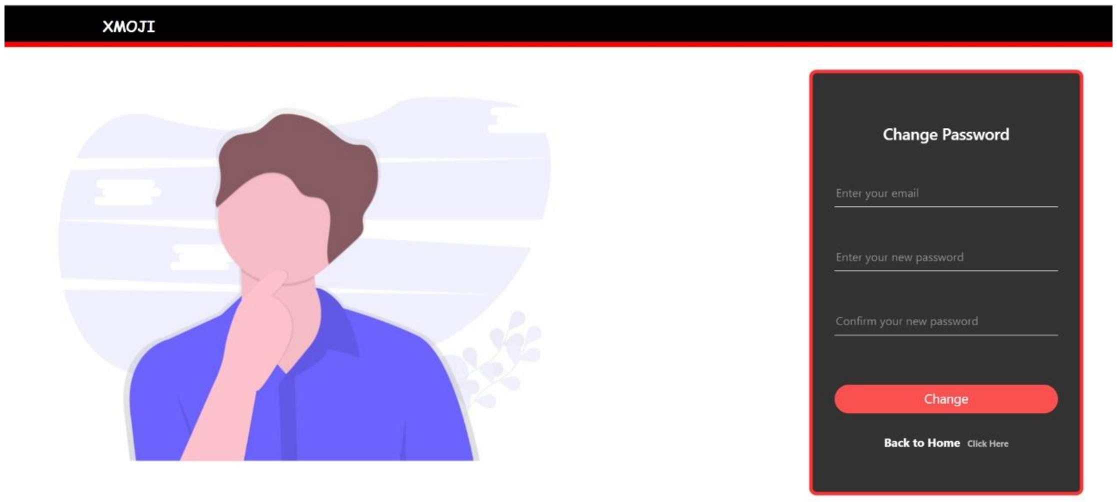
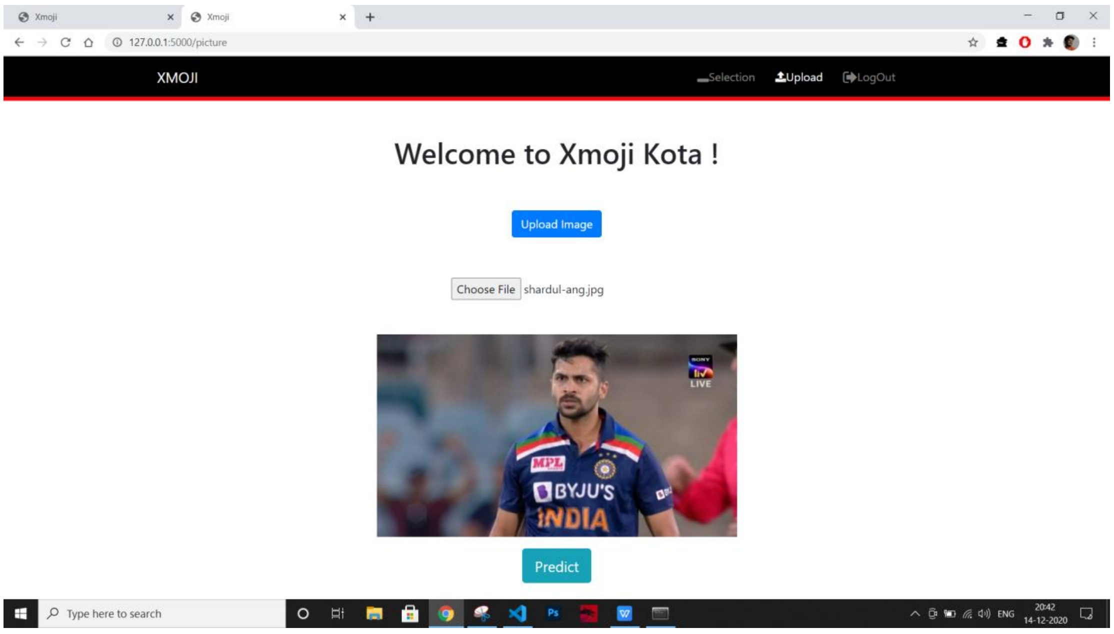


Image Selection for Expression Evaluation:

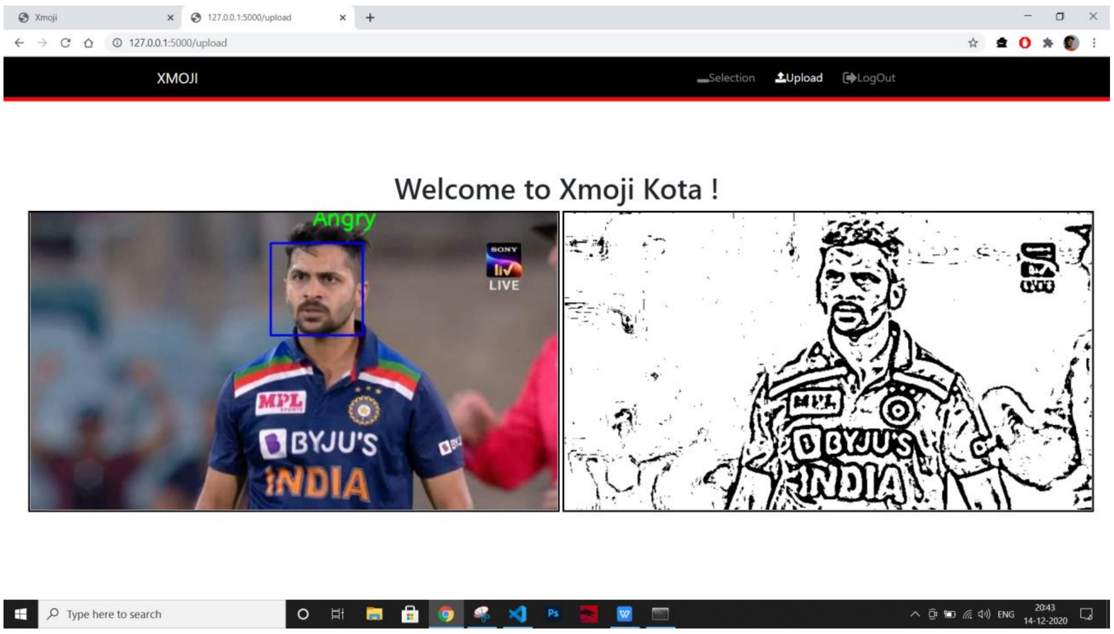


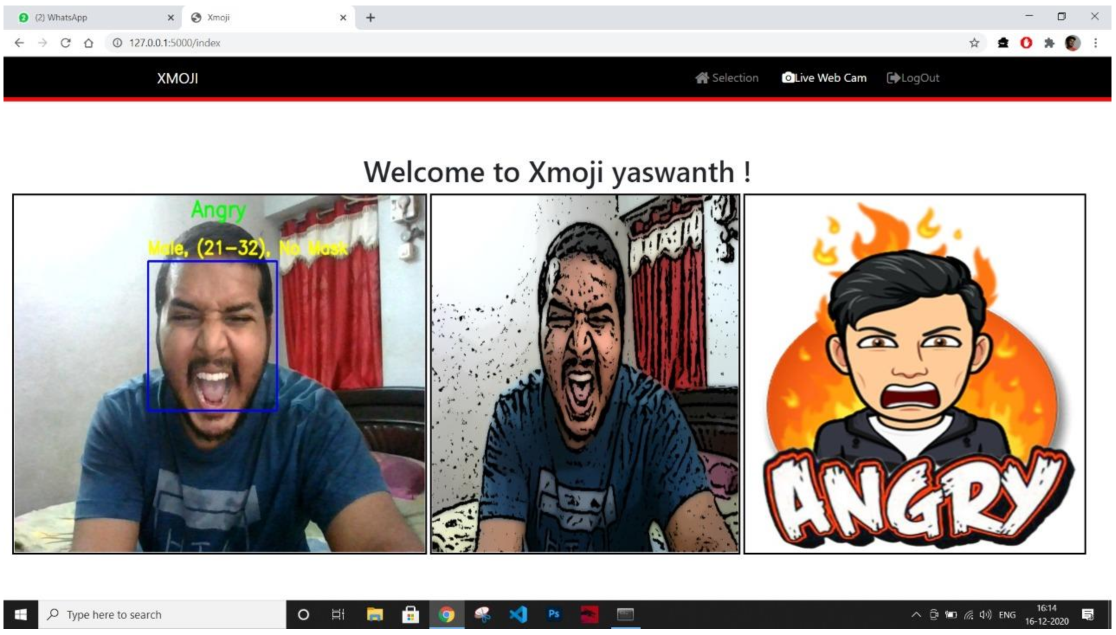
12. TEST CASES

Prediction of Expression:

Angry:

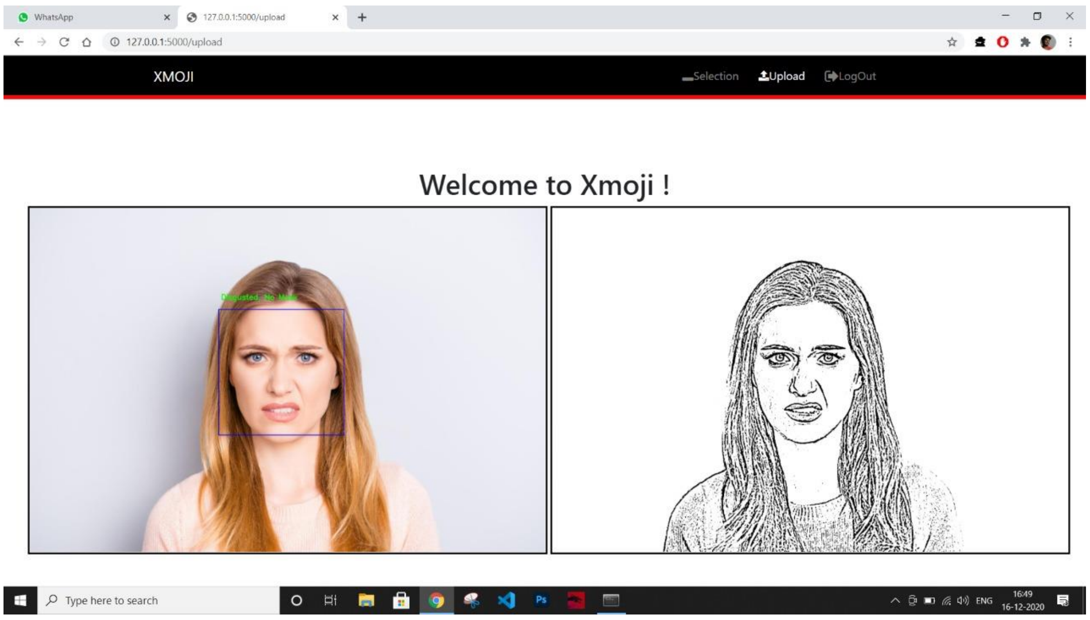
ATTRIBUTES	PREDICTED	ORIGINAL
GENDER	MALE	MALE
MASK	NO	NO
AGE	21-32	23
EMOTION	ANGRY	ANGRY





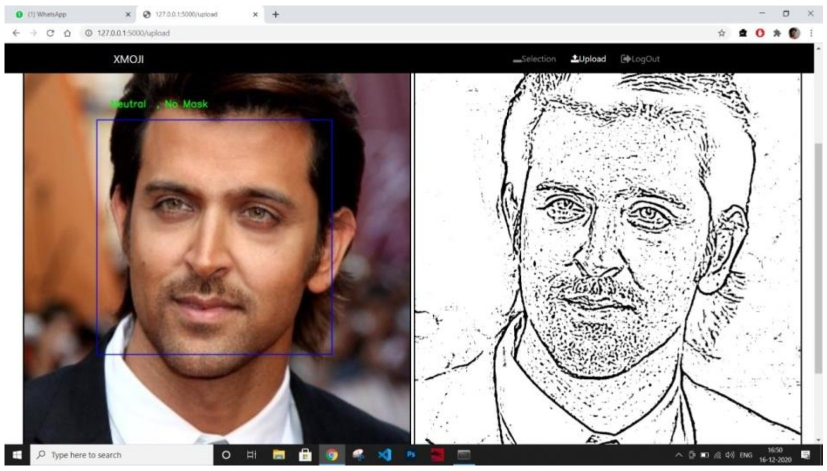
Disgust:-

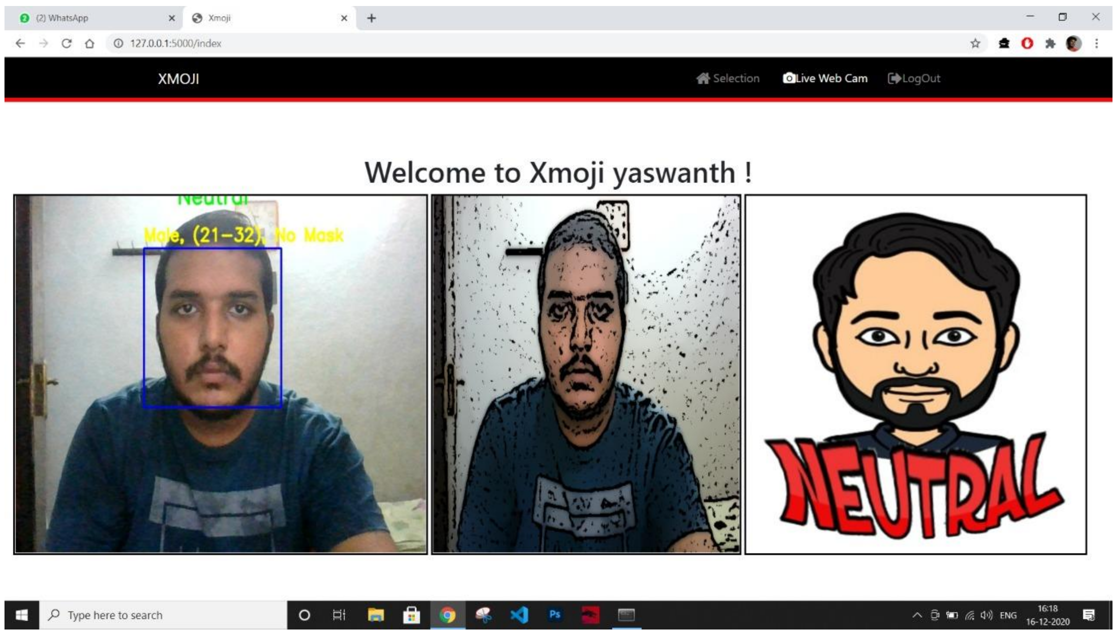
ATTRIBUTES	PREDICTED	ORIGINAL
GENDER	FEMALE	FEMALE
MASK	NO	NO
EMOTION	DISGUST	DISGUST



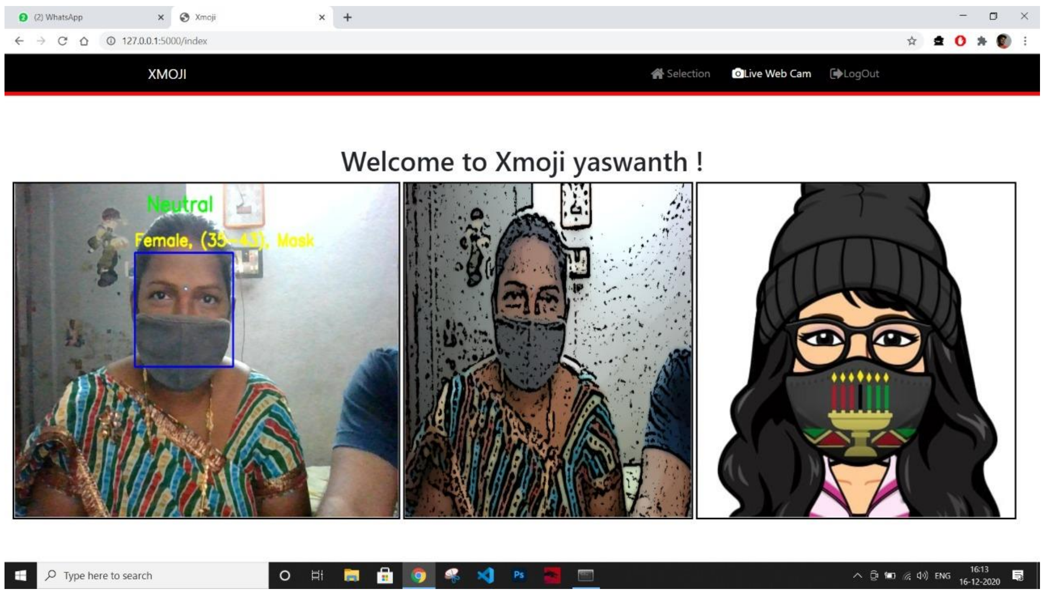
Neutral:

ATTRIBUTES	PREDICTED	ORIGINAL
GENDER	MALE	MALE
MASK	NO	NO
EMOTION	NEUTRAL	NEUTRAL



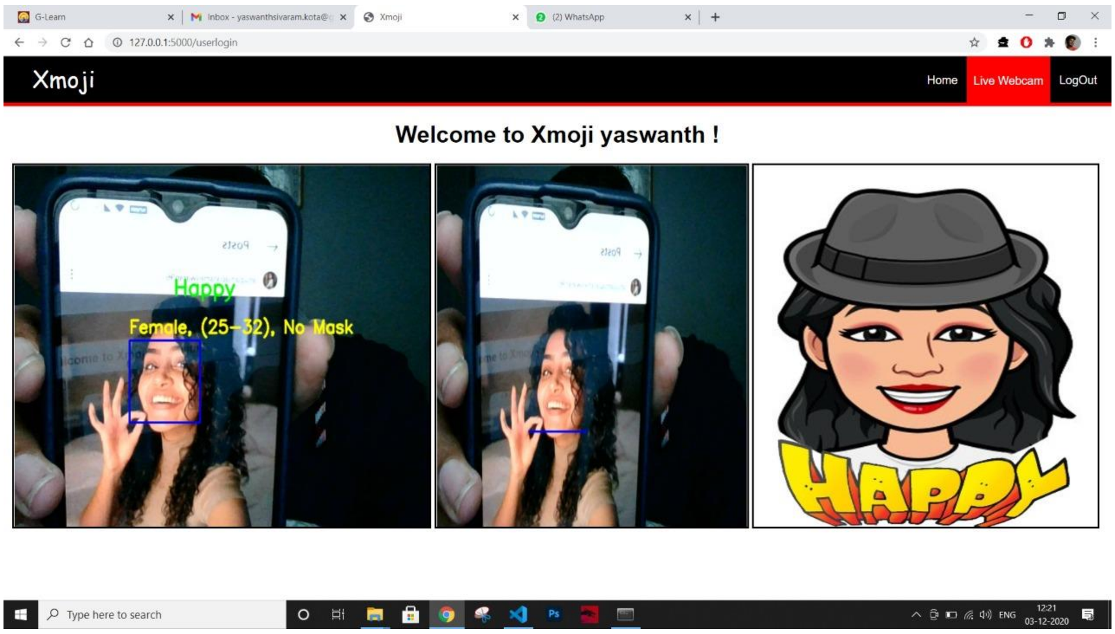


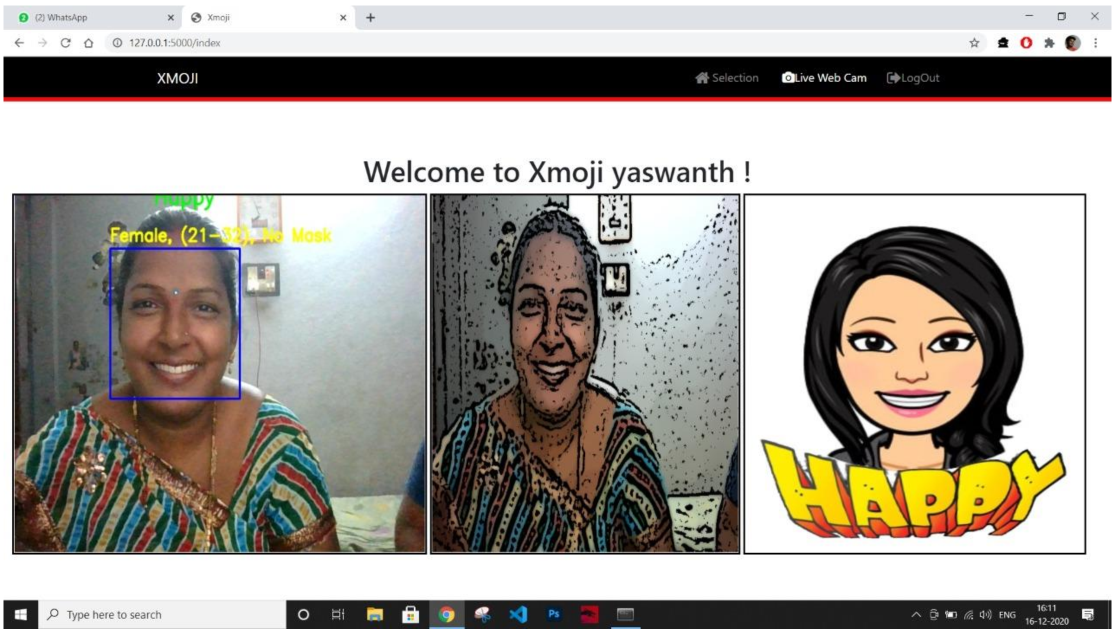
With mask:-



Happy:-

ATTRIBUTES	PREDICTED	ORIGINAL
GENDER	FEMALE	FEMALE
AGE	35-43	40
EMOTION	HAPPY	HAPPY
MASK	NO	NO


127.0.0.1:5000/userlogin
Xmoji
Home
Live Webcam
LogOut
Welcome to Xmoji yaswanth !
Happy
Female, (25-32), No Mask
HAPPY


127.0.0.1:5000/index
XMOJI
Selection
Live Web Cam
LogOut
Welcome to Xmoji yaswanth !
Happy
Female, (21-32), No Mask
HAPPY

13. 1CONCLUSION

This project proposes an approach for recognizing the category of facial expressions and displays the expression in form of emoji regarding to the expression. Face Detection and Extraction of expressions from facial images is useful in many applications, such as robotics vision, chatting/messaging applications, etc…This project’s objective was to develop emojis from facial expression implementing the computer visions and enhancing the advanced feature extraction and classification in face expression recognition.

More efforts should be made to improve the classification performance for important applications.

Our future work will focus on improving the performance of the system and deriving more appropriate classifications which may be useful in many real world applications.

14. FUTURE SCOPE

Emoji generation systems have been occupied a lot over the past years. The focus has definitely shifted from posed expression recognition to spontaneous expression recognition. Promising results can be obtained under face registration errors, fast processing time, and high correct recognition rate (CRR) and significant performance improvements can be obtained in our system. System is fully automatic and has the capability to work with images feed. It is able to recognize spontaneous expressions. Our system can be used in Digital Cameras wherein the image can be captured,when the person wants his emoji according to his expression.

Rooms in homes can set the lights, television to a person's taste when they enter the room regarding to his mood in the cam/emojis.

Our system can be used to detect and track a user's state of mind, and in mini-marts, shopping center to view the feedback of the customers to enhance the business etc…

15. REFERENCES

1)Gender Classification Based on Asian Faces using Deep Learning

https://ieeexplore.ieee.org/document/8906399

2) Gender Recognition from Facial Images using Convolutional Neural Network

https://ieeexplore.ieee.org/document/8985914/citations?tabFilter=papers#citations

3)Facial emotion recognition using deep learning: review and insights “WafaMellouka” ,“WahidaHandouzia”

https://www.sciencedirect.com/science/article/pii/S1877050920318019

4) Automatic Facial Expression Recognition Using DCNN

https://www.sciencedirect.com/science/article/pii/S1877050916314752

5) Predict age and gender using convolutional neural network and opencv

https://www.kdnuggets.com/2019/04/predict-age-gender-using-convolutional-neural-network-opencv.html

6)Face Mask Detection using TensorFlow and OpenCV

https://towardsdatascience.com/covid-19-face-mask-detection-using-tensorflow-and-opencv-702dd833515b

7) Facial Mask Detection using Semantic Segmentation

https://www.researchgate.net/publication/336952877_Facial_Mask_Detection_using_Semantic_Segmentation

8) Gender & Age Classification using OpenCV Deep Learning

https://www.learnopencv.com/age-gender-classification-using-opencv-deep-learning-c-python/