MARRI LAXMAN REDDY GROUP OF INSTITUTIONS

MLR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Affiliated to JNTUH,Approved by AICTE

Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India

NBA NATIONAL BOARD OF ACCREDITATION

NAAC

Department of Information Technology

Mini Project Report

On

"Human Expression Analysis using Machine Learning"

Submitted in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Information Technology

By

D. SWETHAN	**19R21A12D7**
K. HARINI REDDY	**19R21A12F0**
SHREYAS GOPARAPU	**19R21A12G9**
S. SATVIK REDDY	**19R21A12H0**

Under the guidance of

Mr. V. Pavan Kumar

Assistant Professor

MARRI LAXMAN REDDY GROUP OF INSTITUTIONS

MLR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Affiliated to JNTUH,Approved by AICTE

Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India

NBA NATIONAL BOARD OF ACCREDITATION

NAAC

Department of Information Technology

CERTIFICATE

This is to certify that the project entitled **"Human Expression Analysis using Machine Learning"** has been submitted by **D.Swethan(19R21A12D7) K.Harini(19R21A12F0) Shreyas Goparapu(19R21A12G9) S.Satvik Reddy(19R21A12H0)** in the partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Information Technology from Jawaharlal Nehru Technological University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Internal Guide

Mr. V. Pavan Kumar

Head of the Department

Dr. N.V. Rajasekhar Reddy

External Examiner

DECLARATION

We hereby declare that the project entitled **"Human Expression Analysis using Machine Learning"** is the work done during the period from **September 2021 to December 2021** and is submitted in the partial fulfilment of the requirements for the award of degree of Bachelor of technology in Information Technology from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

D. SWETHAN (19R21A12D7)
K. HARINI REDDY (19R21A12F0)
SHREYAS GOPARAPU (19R21A12G9)
S. SATVIK REDDY (19R21A12H0)

ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all.

First of all, we would like to express our deep gratitude towards our internal guide **Mr. V. Pavan Kumar, Assistant Professor, Department of IT** for his support in the completion of our mini project. We wish to express our sincere thanks to **Dr. N. V. RAJASEKHAR, HOD, Department of IT** and also to Principal **Dr.K. SRINIVASA RAO** for providing the facilities to complete the dissertation.

We would like to thank all our faculty, coordinators and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

D. SWETHAN (19R21A12D7)
K. HARINI REDDY (19R21A12F0)
SHREYAS GOPARAPU (19R21A12G9)
S. SATVIK REDDY (19R21A12H0)

ABSTRACT

The aim of Real-time Data Visualization of Facial Expression Data using Keras and Plotly is to detect and classify human facial expressions from image sequence. This is also used to cope with emotional health problems caused due to negativity in our day to day lives. This software program makes use of biometric markers to hit upon feelings in human faces. The six expressions: happiness, sadness, anger, fear, anger and neutral may be detected on the human face.

Given a photo of a person, it recognizes different types of expression of the person by using deep learning techniques & back propagation algorithm and visualizes the types of expression by using different visualization tools.

CONTENTS

Table of content **Page No**

1. INTRODUCTION

The use of technology in society has greatly increased in recent decades. Nowadays, machines are used in many different industries. As their personal exposure grows, communication should also be smooth and natural. To achieve this, machines must be empowered to understand the environment.

When machines are capable of recognizing the environment, it's a far form of system expertise that has improved. People use their senses to benefit perception into their nature. Therefore, system expertise pursuits to imitate human senses as a way to meet their needs. Nowadays, machines have many approaches to seize cameras and sensors. Therefore, making use of these facts with suitable algorithms permits to generate system expertise.

The intention of this program is to enhance the Automatic Facial Expression Recognition System that may take pics of a person's face that carries different expressions which includes insert and spot once more divide it into six classes which includes - I. Neutral, II. Angry, III. Fear, IV. Happy, V. Sad, VI. Surprise.

1.1 Objective

The objective of this project is to develop an algorithm to recognize the expression of human face accurately and to visualize the expression into six universal emotions using different visualization techniques. The recent years have seen a rise in the number of papers published that use deep learning for facial emotion recognition. These papers used freely available datasets with state of art models achieving an accuracy of 0.66. With this in mind, a number of different models both new and old will be experimented with to arrive at a final model with comparable results

1.2 Existing System

The existing methods for human face detection include CV and Tensorflow, Fractional Max Pooling, Le Net. These use CNN (Convolutional Neural Networks).

Disadvantages of existing system

One of the biggest disadvantages of the previously existing systems is its accuracy. As seen in the graph, the highest accuracy achieved in the methods considered was around 64%. Also, the biggest challenge is to differentiate between the expressions ‘Sad’ and ‘Fear’. The existing models run into ambiguity when the above two expressions are in question.

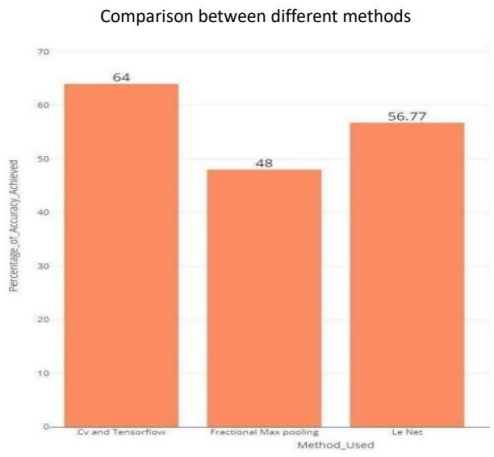


Figure 1.2.1

1.3 Proposed System

The proposed system will train the dataset using back propagation algorithm to give a model that detects the expressions in the input picture. Now, using Plotly we aim to visualize the output in form of graphs. Also, it will detect whether the emotional index of negative expressions like anger and sadness is highest. If yes, using python to display some message to help in making the user feel better.

2. LITERATURE SURVEY

1. Automatic facial expression recognition based on a deep convolutional-neural-network structure (2017)

Shan, Ke, Junqi Guo, Wenwan You, Di Lu, and Rongfang Bie

The authors used a deep convolutional neural network (CNN) to develop a facial recognition system that could achieve automatic recognition of facial features. The research was done in four stages, input, processing, recognition and output. The two datasets used are JAFFE which has 213 images of Japanese women and CK+ which has 593 grayscale images. Haar-like feature in OpenCV is used for detecting the expressions. To make the results more convincing, they were compared to the ones obtained by using KNN algorithm. The accuracy of this system reached 76.7% and 80.303% for JAFFE and CK+ respectively. However it has been observed that it is easier to achieve maximum accuracy for small datasets. The main challenge is to obtain accuracy for large datasets as more the number of images, more will be the accuracy of the trained model.

2. Going deeper in facial expression recognition using deep neural networks (2016)

Mollahosseini, Ali, David Chan, and Mohammad H. Mahoor

The method proposed is a new deep neural network architecture that consists of two convolutional layers for facial expression recognition. These layers are followed by max pooling and four inception layers. To validate the accuracy of this method the authors used seven databases. This method is considered to be subject independent as well as cross database due to the large number of training and test databases.

3. Facial expression recognition based on deep evolutional spatial-temporal networks (2017)

Zhang, Kaihao, Yongzhen Huang, Yong Du, and Liang Wang

This paper proposed a part-based hierarchical bidirectional recurrent neural network (PHRNN) to detect facial expressions in temporal sequences and a multi-signal convolutional neural network (MSCNN) to extract spatial features from the still images of the video frames. This method is effective s the combination of these networks boosts the extraction of facial features by extracting the dynamic as well as still information of the frames. This combined method was applied on three datasets namely CK+, Oulu-CASIA and MMI and it was observed that it outperforms the previously introduced methods by reducing errors by great margins.

3. SYSTEM REQUIREMENTS

3.1. Hardware Requirements

Processor : Intel Core i3

RAM : 4GB

CPU : 2 GHZ or faster

Hard Disk : Minimum 500 GB

3.2 Software Requirements:

Operating System : Windows 7 onwards

Programming Language : Python 3.8

Supporting libraries : Tensorflow / keras, plotly, skimage

4. SYSTEM DESIGN

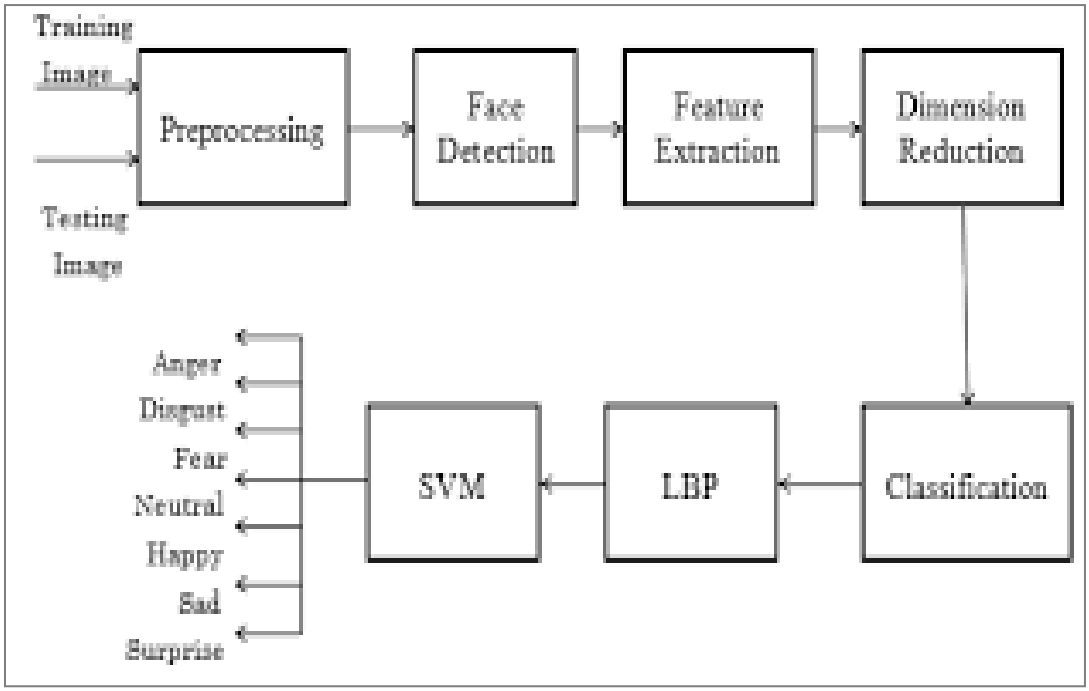


Figure 4.1 ***Steps for face expression recognition***

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is centered and occupies about the same amount of space in each image. The training set consists of 28,709 examples. The public test set used for the leaderboard consists of 3,589 examples. The final test set, consists of another 3,589 examples. This dataset was prepared by Pierre-Luc Carrier and Aaron Courville, as part of a research project.

4.1 Dataflow Diagram

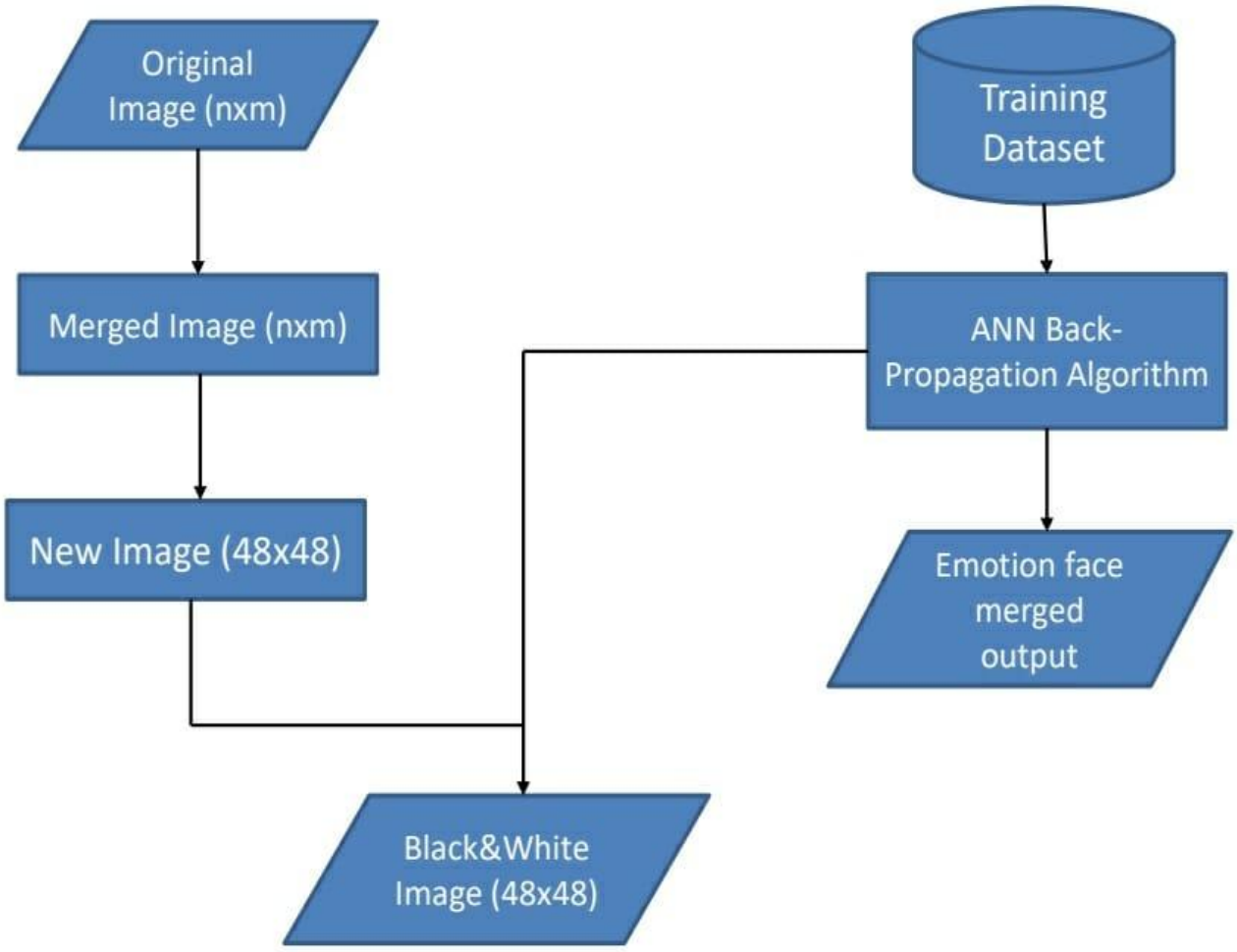


Figure 5.1 Flowchart

5. IMPLEMENTATION

5.1 Coding

Training the dataset:

```
from google.colab import drive
drive.mount('/content/drive')

import tensorflow as tf

import keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, AveragePooling2D
from keras.layers import Dense, Activation, Dropout, Flatten

from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator

import numpy as np
import matplotlib.pyplot as plt

num_classes = 7 #angry, disgust, fear, happy, sad, surprise, neutral
batch_size = 128
epochs = 50

with open("./drive/My Drive/fer2013.csv") as f:
 content = f.readlines()

lines = np.array(content)

num_of_instances = lines.size
print("number of instances: ",num_of_instances)
print("instance length: ",len(lines[1].split(",")[1].split(" ")))

x_train, y_train, x_test, y_test = [], [], [], []
```

```
for i in range(1,num_of_instances):
  try:
    emotion, img, usage = lines[i].split(",")
    val = img.split(" ")
    pixels = np.array(val, 'float32')
    emotion = keras.utils.to_categorical(emotion, num_classes)
    if 'Training' in usage:
      y_train.append(emotion)
      x_train.append(pixels)
    elif 'PublicTest' in usage:
      y_test.append(emotion)
      x_test.append(pixels)
  except:
    print("", end="")

x_train = np.array(x_train, 'float32')
y_train = np.array(y_train, 'float32')
x_test = np.array(x_test, 'float32')
y_test = np.array(y_test, 'float32')

x_train /= 255 #normalize inputs between [0, 1]
x_test /= 255

x_train = x_train.reshape(x_train.shape[0], 48, 48, 1)
x_train = x_train.astype('float32')
x_test = x_test.reshape(x_test.shape[0], 48, 48, 1)
x_test = x_test.astype('float32')

print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')

#construct CNN structure
model = Sequential()

#1st convolution layer
model.add(Conv2D(64, (5, 5), activation='relu', input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(5,5), strides=(2, 2)))
```

```
#2nd convolution layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

#3rd convolution layer
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))

model.add(Flatten())

#fully connected neural networks
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(num_classes, activation='softmax'))
#------------------------------
#batch process
gen = ImageDataGenerator()
train_generator = gen.flow(x_train, y_train, batch_size=batch_size)

model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.Adam(),
metrics=['accuracy'])
fit = True

if fit == True:
  model.fit_generator(train_generator, steps_per_epoch=batch_size, epochs=epochs)
else:
  model.load_weights('/data/facial_expression_model_weights.h5') #load weights

def emotion_analysis(emotions):
 objects = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')
 y_pos = np.arange(len(objects))

 plt.bar(y_pos, emotions, align='center', alpha=0.5)
 plt.xticks(y_pos, objects)
 plt.ylabel('percentage')
```

```
 plt.title('emotion')

 plt.show()

model_json = model.to_json()
with open("model.json", "w") as json_file:
 json_file.write(model_json)
model.save_weights("model.h5")

#Saving the model
model.save('model.h5')
#Evaluation
train_score = model.evaluate(x_train, y_train, verbose=0)
print('Train loss:', train_score[0])
print('Train accuracy:', 100*train_score[1])

test_score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', test_score[0])
print('Test accuracy:', 100*test_score[1])

!ls model.h5

!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)

model.save('model.h5')
model_file = drive.CreateFile({'title' : 'model.h5'})
model_file.SetContentFile('model.h5')
model_file.Upload()

# download to google drive
drive.CreateFile({'id': model_file.get('id')})
```

Visualising the analysis:

```
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator

import os

import plotly
import numpy as np
#import tensorflow
from keras.models import load_model
import plotly.graph_objects as go
import plotly.offline as plo
from plotly import subplots
from skimage import io
import tkinter
import random

model = load_model('model.h5') #trained dataset model
def emotion_analysis(emotions):
    objects = ('Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral')
    y_pos = np.arange(len(objects))
    fig1 = go.Bar(x=objects, y=emotions*100, marker={'color': 'crimson'}, showlegend=False,
name="")

    fig2 = go.Pie(labels=objects, values=emotions*100, name="")

    fig3 = go.Funnel(y=objects,x=emotions*100,name="",marker={'color' : 'tan'},
showlegend=False)

    fig4 = go.Scatter(x=objects, y=emotions*100, name="",marker={'color' : 'teal'},
showlegend=False, fill= 'tonexty', fillcolor='rgb(111, 231, 219)')

    fig5 = go.Scatter(x=objects, y=emotions*100, name="",mode= 'markers', marker={'color' :
'yellow', 'size' : emotions*150}, showlegend=False)

    img = io.imread("10.jpeg") #reading the image from the folder for
```

```
    fig6 = go.Image(z=img)

    figure = subplots.make_subplots(
    rows=2,
    cols=3,
    specs=[[{"type": "bar"}, {"type": "scatter"}, {"type": "pie"}],
           [{"type": "scatter"}, {"type": "funnel"}, {"type": "image"}] ],
    subplot_titles= ("Visualisation through Bar Graph","Visualisation through Bubble
Chart","Visualisation through Pie Chart",
                     "Visualisation through Area Chart","Visualisation through Funnel Chart","Image
for Emotion Recognition")
    )

    figure.add_trace(fig1, 1, 1)
    figure.add_trace(fig5, 1, 2)
    figure.add_trace(fig2, 1, 3)
    figure.add_trace(fig4, 2, 1)
    figure.add_trace(fig3, 2, 2)
    figure.add_trace(fig6, 2, 3)
    figure.update_layout(
        {
            "autosize" : True,
            "title" :{"text" : "Data Visualisation for Emotion Recognition", "font" : {"size" : 30}},
            "xaxis_title" : {"text" : "Emotion", "font" : {"size" : 20}},
            "yaxis_title" : {"text" : "Percentage", "font" : {"size" : 20}},
            "template" : "plotly_dark"
        }
    )
    figure.show() #to show the output on renderer
    plotly.offline.plot(figure, filename='file.html') #to show it as a html page in a browser

file = '10.jpeg' #image taken for the analysis
true_image = image.load_img(file)
img = image.load_img(file, color_mode = "grayscale", target_size=(48, 48))

x = image.img_to_array(img)
x = np.expand_dims(x, axis = 0)

x /= 255 #normalising pixels in the image array
```

```
happy_message = {
   1 : "Hey! Keep smiling.",
   2 : "A smile is curve that can set everything straight! Keep smiling ",
   3 : "Keep smiling! It’s free therapy !",
   4 : "You’ll find that life is still worthwile, if you just smiling!",
   6 : "A smile is happiness that you’ll find right under your nose! Keep smiling",
   7 : "Hoping that you always find a reason to smile ",
   8 : "Life is a camera, so keep smiling!",
   9 : "Always smile, because your smile is reason for others to smile! ",
   10 : "Life is short! Smile while you still have teeth ",
}

angry_message = {
   1 : "Now the sky is overcast, but soon the sun will appear. The problems do not last forever, everything will be fine!",
   2 : "Every time you feel upset, remember that God does not give us trials that we cannot withstand.",
   3 : "“Don’t give up when dark times come. The more storms you face in life, the stronger you’ll be. Hold on. Your greater is coming.”",
   4 : "Cheer up, my dear. After every storm comes the sun. Happiness is waiting for you ahead.",
   5 : "If it comes, let it come. If it goes, it's ok, let it go. Let things come and go. Stay calm, don't let anything disturb your peace, and carry on",
   6 : "Getting angry in a stressful situation is like trying to clean something with dirt",
   7 : "Free your hearts of anxiety, pain and anger, to have peace within your heart and soul",
   8 : "Reacting in anger or annoyance will not advance one's ability to persuade.",
   9 : "When one burns one's bridges, what a very nice fire it makes",
   10 : "Calm mind brings inner strength and self-confidence, so that’s very important for good health.",
}

fear_message = {
   1 : "Courage is knowing what not to fear.",
   2 : "“Fears are educated into us, and can, if we wish, be educated out.”",
   3 : "“Curiosity will conquer fear even more than bravery will.”",
   4 : "“Fear: False Evidence Appearing Real.”",
   5 : "“I am not afraid of tomorrow, for I have seen yesterday and I love today.”— William Allen White",
   6 : "“Laughter is poison to fear, so cheer up my friend”",
   7 : "“Fear is only as deep as the mind allows.”",
```

8 : "“One of the greatest discoveries a man makes, one of his great surprises, is to find he can do what he was afraid he couldn’t do, so don’t stop trying”",
9 : "The only thing we have to fear is fear itself.",
10 : "“To overcome fear, here’s all you have to do: realize the fear is there, and do the action you fear anyway.”",
}

surprise_message = {
1 : "Oh you seem surprised! Hope it was something good □",
2 : "HEY! Hope you liked the surprise!",
3 : "Oh definitely share that happiness you got when you were surprised!",
4 : "Hey! Please do share the story of your surprised reaction!",
5 : "Haha who knew! Hope you like that surprise!",
6 : "Your expression tells that you were surprised by something! Hope it was good",
7 : "Gotcha! ",
8 : "Unbelievable! You never saw it coming did you?!",
9 : "That caught you off guard! ",
10 : "You seem to be rooted to the spot! ",
}

neutral_message = {
1 : "Here let me tell you a joke: What shoes do bears wear? They don't, they go bear feet.",
2 : "What kind of clothes do houses wear? Adress.",
3 : "Hey let’s put a smile on that face! ",
4 : "Here’s a joke: I forgot how to throw a boomerang, but it came back to me.",
5 : "There’s always a reason to smile! Find it □",
6 : "Why don't skeletons watch scary movies? Because they don't have the guts",
7 : "What you call an owl that does magic? HOO-Dini ",
8 : "How do you call a group of unorganized cats? A cat-astrophe.",
9 : "Life is short! Smile while you still have teeth",
10 : "What's barber's favorite instrument? A hair-monica.",
}

sad_message = {
1 : "And just like any other hard time, you’ll make it through this one too. Life is tough but so are you!",
2 : "Life is tough but so are you!",
3 : "Everything is going to work out just fine",
4 : "Sad Message 4",
5 : "Hey life is a gift, don’t waste it in melancholy",

```
    6 : "Just fight a little longer my friend, it’s all worth it in the end. ",
    7 : "Be proud of how hard you are trying",
    8 : "Hang in there, it’s astonishing how short a time wonderful things take to happen.",
    9 : "What did tomato say to the other tomato during a race? Ketchup.",
    10 : "Why did the bicycle fall over? Because it was two tired.",
}

disgust_message = {
    1 : "Hey please don’t waste your time on being disgusted by something as pety!",
    2 : "Relish everything that's inside of you, the imperfections, the darkness, the richness and
light and everything. And that makes for a full life.",
    3 : "Live daringly, boldly, fearlessly. Taste the relish to be found in competition - in having
put forth the best within you.",
    4 : "Look forward to your day! Let bygones be bygones",
    5 : "Discuss the situation with someone and you’ll find yourself smiling in no time.",
    6 : "Take control of that disgust! Don't let it lead your day or life.",
    7 : "Try not to be disgusted with people, it can lead to grave social consequences.",
    8 : "Don’t let the disgust engulf your thinking! Move on!",
    9 : "Hey be optimistic and be productive.",
    10 : "Expose yourself to that object of aversion, to the point where it doesn’t bother you.
Face it confidently! ",
}

custom = model.predict(x)
expression = list(custom[0]*100)
emotion_analysis(custom[0])

root = tkinter.Tk()
root.title("Message")
root.geometry("300x200")
var = tkinter.StringVar()
label = tkinter.Message( root, textvariable=var, relief= tkinter.RAISED, width= 300 )

# ('Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral')

if expression.index(max(expression)) == 0:
    var.set("Angry : " + angry_message[random.randint(1, 10)])
elif expression.index(max(expression)) == 1:
    var.set("Disgust : " + disgust_message[random.randint(1, 10)])
elif expression.index(max(expression)) == 2:
```

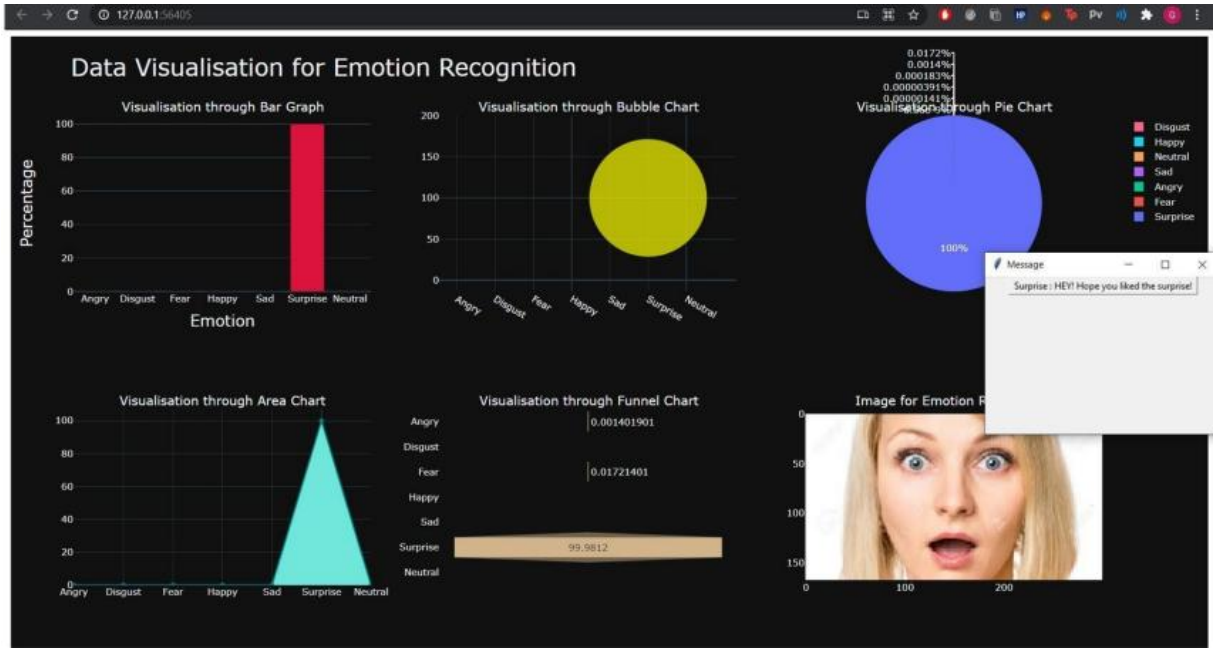
```
    var.set("Fear : " + fear_message[random.randint(1, 10)])
elif expression.index(max(expression)) == 3:
    var.set("Happy : " + happy_message[random.randint(1, 10)])
elif expression.index(max(expression)) == 4:
    var.set("Sad : " + sad_message[random.randint(1, 10)])
elif expression.index(max(expression)) == 5:
    var.set("Surprise : " + surprise_message[random.randint(1, 10)])
elif expression.index(max(expression)) == 6:
    var.set("Neutral : " + neutral_message[random.randint(1, 10)])

label.pack()
root.mainloop()
```

6. RESULT AND DISCUSSION

Following figures show the result of the model:-

Surprise:

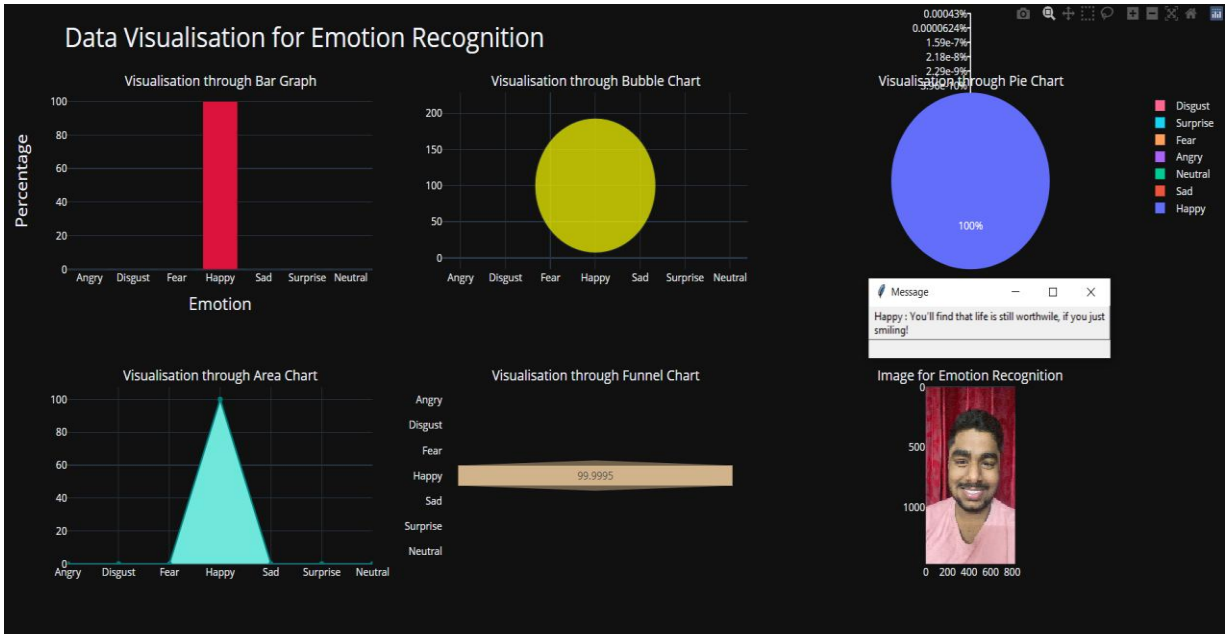
Image 1 – Sample Input



Output

Happy:

Image 2 – Sample Input



Output

Neutral:

Image 3 – Sample Input

Output

On training the FER2013 dataset using the backpropagation algorithm, we were successfully able to differentiate between the six basic expressions happy, sad, angry, surprised, fear and neutral with an accuracy of 71.38%. It was observed that the size of the dataset used to train for the model is directly proportional to accuracy of the results. The accuracy of the results with which an expression can be detected successfully is also dependent on the number of epochs in total while training the dataset.

7. ADVANTAGES AND APPLICATIONS

The accuracy of the proposed system is higher than the previous existing systems. The scope for this program is diversed as it can be used in forensics, social quotient of public, emotional well being, feedback assurance etc. The below figure shows the accuracy among various systems.

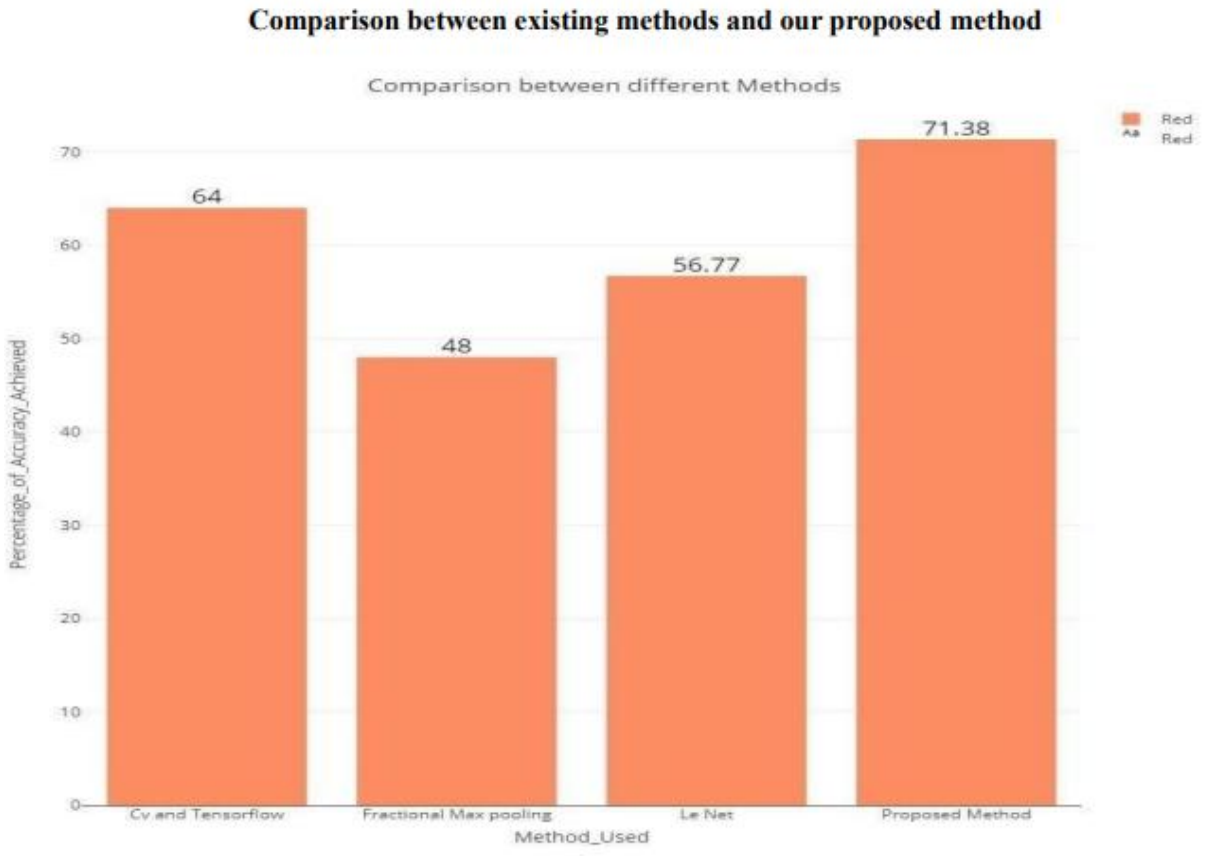


Figure 7.1

9. CONCLUSION

Face expression recognition systems have improved tons over the past decade. The main target has definitely shifted from posed expression recognition to spontaneous expression recognition. Subsequent decades are going to be interesting since robust spontaneous expression recognizers are going to be developed and deployed in real-time systems and utilized in building emotion sensitive HCI interfaces. This is often getting to have an impression on our day to day life by enhancing the way we interact with computers or generally, our surrounding living and work spaces. Having improved techniques to deal with expression variation, within the future it's going to be investigated in additional depth about the face classification problem and optimal fusion of color and depth information. Further study can be laid down within the direction of allele of gene matching to the geometric factors of the facial expressions.

10. BIBLOGRAPHY

[1] Shan, K., Guo, J., You, W., Lu, D., & Bie, R. (2017, June). Automatic facial expression recognition based on a deep convolutional-neural-network structure. In *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)* (pp. 123-128). IEEE.

[2] Uddin, M. Z., Hassan, M. M., Almogren, A., Alamri, A., Alrubaian, M., & Fortino, G. (2017). Facial expression recognition utilizing local direction-based robust features and deep belief network. *IEEE Access*, 5, 4525-4536.

[3] Ding, Y., Zhao, Q., Li, B., & Yuan, X. (2017). Facial expression recognition from image sequence based on LBP and Taylor expansion. *IEEE Access*, 5, 19409-19419.

[4] Zhang, K., Huang, Y., Du, Y., & Wang, L. (2017). Facial expression recognition based on deep evolutional spatial-temporal networks. *IEEE Transactions on Image Processing*, 26(9), 4193-4203.