RAMAIAH
Institute of Technology

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

RAMAIAH INSTITUTE OF TECHNOLOGY
(AUTONOMOUS INSTITUTE AFFILIATED TO VTU)
M. S. R. I. T. POST, BANGALORE – 560054
2021-2022

A report submitted in partial fulfilment of the requirements

of Other Component

Programming Assignment

for the subject

Computer Vision (ISE555)

In

Fifth Semester

Submitted By:

1. ABHISHEK K-1MS20IS003
2. MITHUN R-1MS20IS072
3. MANOJ S-1MS20IS070
4. RUSHAB KESHAV-1MS20IS057

Submitted To:

Dr.Megha P Arakeri
Associate Professor,
Dept. of ISE, RIT

INDEX

ABSTRACT

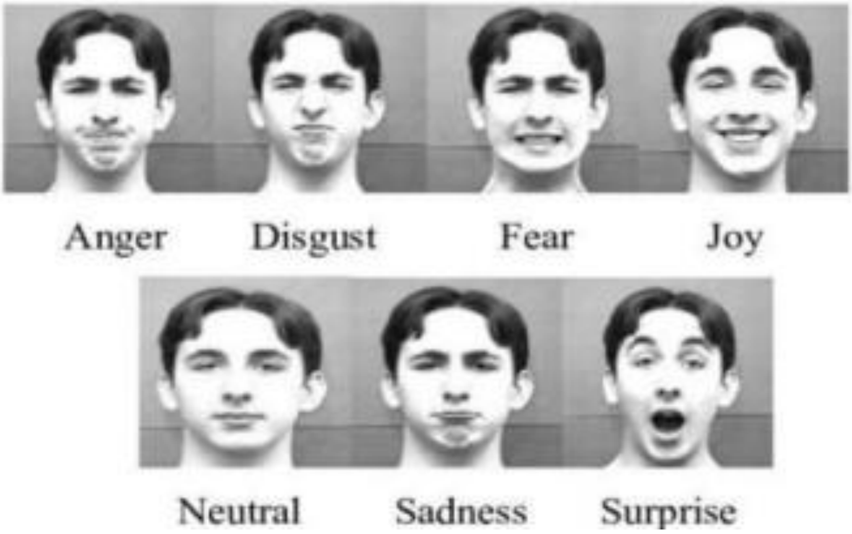
Detecting and recognizing human emotion is not an easy thing to do. This was a big challenge. It involves object detection first and Emotion detection. In this report will use CNN (Convolution Neural Network) model to recognize human emotion from the live feed using the FER2013 (Facial Expressions Recognition 2013) dataset to train the model. The model delivers 54 percent accuracy from recognizing seven basic human emotions such as Neutral, Angry, Disgust, Fear Happy and Sad.

Introduction

The human face is the most important part of a human's body that is used as a container for our five senses such as eyes, ear, tongue, nose, and skin, it is also unique and could be used to identify a person.

The human face is used to describe their emotions through their facial expressions like the movements of the eyes, eyebrows, lips, and mouths. Different activities bring different aspects of emotions and contain complementary information.

Face muscles can produce many face expression configurations. There are seven basic emotions in general such as disgust, angry, fear, sad, happy, and surprise.



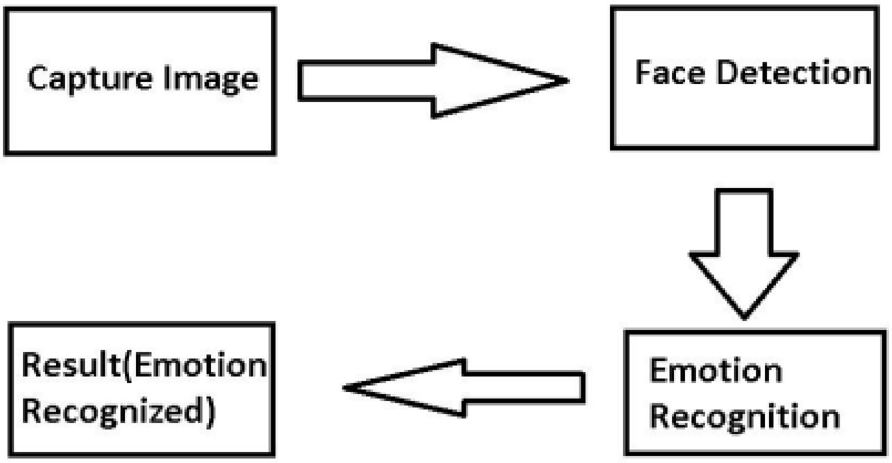
Our goal is to recognize perceived human emotion using a Convolutional Neural Network to create an intelligent face recognition system that can do the same task as humans.

This will be a big challenge in computer vision because many factors have to be taken while developing an emotion recognition system like facial hair, the presence of spectacles, changes in surrounding light, and different illumination. Even though a human can change their behaviour when interacting with others depending on their own perceived emotions

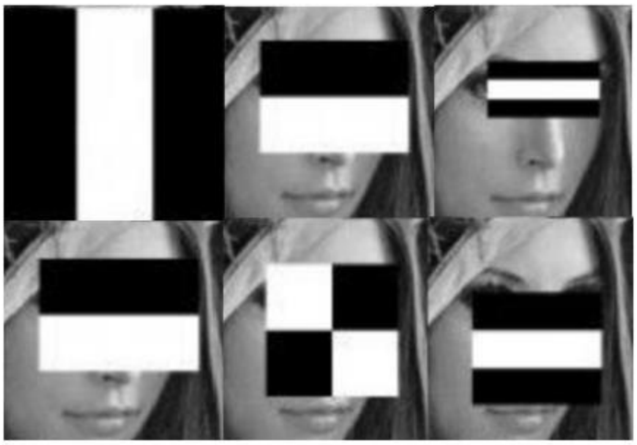
Communication mostly takes place through emotion and has a very important role in interacting with another human. Other than logical intelligence, emotional intelligence is also considered a very important part of human intelligence.

Methodology

This phase will explain how a machine can detect the facial structure of a person and analyse the emotions their express.



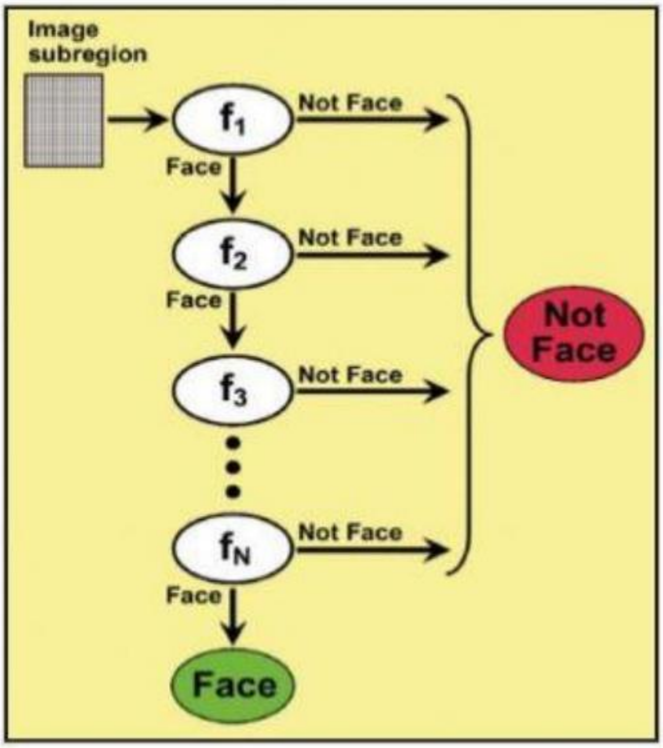
Face Detection: The first phase of this overall methodology is to detect the face from the input images using Viola-Jones algorithm. Before detecting a face, the image is converted into grayscale to provide lesser data process The first step of this algorithm is selecting Haar-like features that consists of one high and one low interval value referred to as the area of light and dark area in two-dimension image



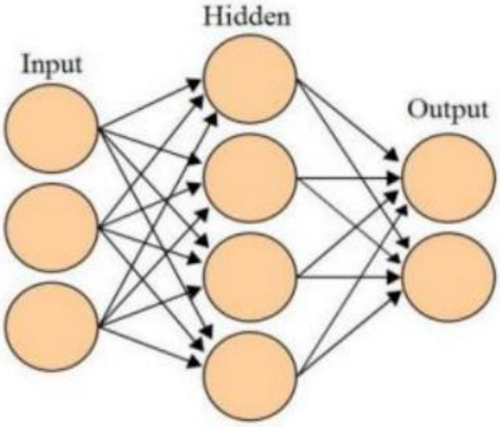
This feature has advantages in the form of a very high computational performance. In human faces there a few key points were used to identify the emotions and classify them because facial structures like eyebrows or the bridge between both the eyes contribute to the detecting algorithm. This means that only a few of these features are important to identify a face.

We use the Adaptive Boosting algorithm to identify the best features to decide the type and size of a feature that goes into the final classifier.

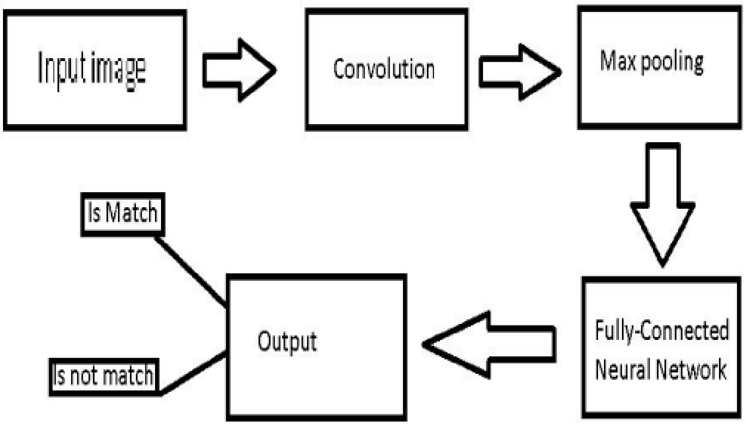
After a few key points are determined, the next phase is the Cascade Classifier. Cascade Classifiers an effective method of object detection to reject undetected areas of the image face by using the classifier that has been trained by the Adaptive Boosting algorithm on each level of classification.



B. **Emotion Recognition using neural network**: The concept of neural networks is a biologically inspired paradigm that allows a computer to learn from the given data. Artificial Neural Network consists of output and input layers, and hidden layers that process input from an input layer into something that the output layer can accept.



Convolutional neural network is a type of neural network that is commonly used in image data, that can be used to detect and recognize objects in an image.



Convolutional Neural Network uses the convolution process by moving a convolutional kernel of a certain size to an image. After that the computer gets new representative information from the multiplication of parts of the image with the filter used.

The first step of the algorithm is to split the image into overlapping smaller images and then each smaller image will be the input which will be fed into the neural network to produce a feature representation. Hence CNN can recognize an object, wherever it appears in the image.

After that, the results will be down sampled, which is useful for retrieving the largest pixel in each kernel pool. In that way, even when reducing the number of parameters, the most important information from that section is still retrieved. Then the final process is the fully connected layer that is used to decide whether the images match or not.

Implementation

```
# import required packages
import cv2
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense, Dropout, Flatten
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator

# Initialize image data generator with rescaling
train_data_gen = ImageDataGenerator(rescale=1./255)
validation_data_gen = ImageDataGenerator(rescale=1./255)

# Preprocess all test images
train_generator = train_data_gen.flow_from_directory(
        'data/train',
        target_size=(48, 48),
        batch_size=64,
        color_mode="grayscale",
        class_mode='categorical')

# Preprocess all train images
validation_generator = validation_data_gen.flow_from_directory(
        'data/test',
        target_size=(48, 48),
        batch_size=64,
        color_mode="grayscale",
        class_mode='categorical')

# create model structure
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_model.compile(loss='categorical_crossentropy',
optimizer=Adam(lr=0.0001, decay=1e-6), metrics=['accuracy'])

# Train the neural network/model
```

```
emotion_model_info = emotion_model.fit_generator(
        train_generator,
        steps_per_epoch=28709 // 64,
        epochs=50,
        validation_data=validation_generator,
        validation_steps=7178 // 64)

# save model structure in jason file
model_json = emotion_model.to_json()
with open("emotion_model.json", "w") as json_file:
    json_file.write(model_json)

# save trained model weight in .h5 file
emotion_model.save_weights('emotion_model.h5')

import cv2
import numpy as np
from keras.models import model_from_json



emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral", 5: "Sad", 6: "Surprised"}

# load json and create model
json_file = open('model/emotion_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
emotion_model = model_from_json(loaded_model_json)
```

```
# load weights into new model
emotion_model.load_weights("model/emotion_model.h5")
print("Loaded model from disk")

# start the webcam feed
#cap = cv2.VideoCapture(0)

# pass here your video path
# you may download one from here : https://www.pexels.com/video/three-girls-laughing-5273028/
cap = cv2.VideoCapture("C:\\JustDoIt\\ML\\Sample_videos\\emotion_sample6.mp4")

while True:
    # Find haar cascade to draw bounding box around face
    ret, frame = cap.read()
    frame = cv2.resize(frame, (1280, 720))
    if not ret:
        break
    face_detector = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_default.xml')
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # detect faces available on camera
    num_faces = face_detector.detectMultiScale(gray_frame, scaleFactor=1.3, minNeighbors=5)

    # take each face available on the camera and Preprocess it
    for (x, y, w, h) in num_faces:
        cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (0, 255, 0), 4)
        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)

        # predict the emotions
        emotion_prediction = emotion_model.predict(cropped_img)
        maxindex = int(np.argmax(emotion_prediction))
        cv2.putText(frame, emotion_dict[maxindex], (x+5, y-20), 
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2, cv2.LINE_AA)

    cv2.imshow('Emotion Detection', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

import numpy as np
from keras.models import model_from_json
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix, 
classification_report,ConfusionMatrixDisplay



emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: 
"Neutral", 5: "Sad", 6: "Surprised"}
```

```
# load json and create model
json_file = open('model/emotion_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
emotion_model = model_from_json(loaded_model_json)

# load weights into new model
emotion_model.load_weights("model/emotion_model.h5")
print("Loaded model from disk")

# Initialize image data generator with rescaling
test_data_gen = ImageDataGenerator(rescale=1./255)

# Preprocess all test images
test_generator = test_data_gen.flow_from_directory(
        'data/test',
        target_size=(48, 48),
        batch_size=64,
        color_mode="grayscale",
        class_mode='categorical')

# do prediction on test data
predictions = emotion_model.predict_generator(test_generator)

# see predictions
# for result in predictions:
```

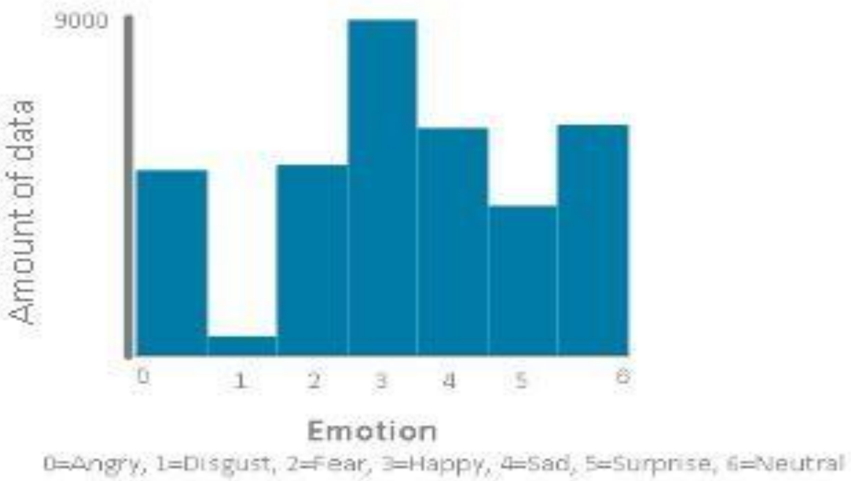
```
#     max_index = int(np.argmax(result))
#     print(emotion_dict[max_index])


print("------------------------------------------------------------------------------------")
# confusion matrix
c_matrix = confusion_matrix(test_generator.classes, predictions.argmax(axis=1))
print(c_matrix)
cm_display = ConfusionMatrixDisplay(confusion_matrix=c_matrix, display_labels=emotion_dict)
cm_display.plot(cmap=plt.cm.Blues)
plt.show()


# Classification report
print("------------------------------------------------------------------------------------")
print(classification_report(test_generator.classes, predictions.argmax(axis=1)))
```

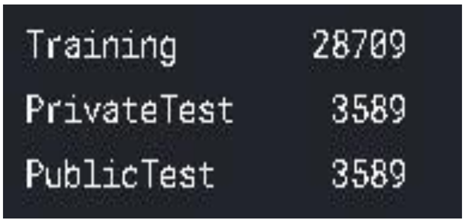
Results

The first step is to create a conventional neural network model that is used to train facial emotion. Before training the model, the first thing is to prepare the dataset. The dataset that used is Facial Expressions Recognition-2013 (FER-2013) because this is a challenging dataset that contains 35.887 images that are not aligned.



FER-2013 dataset with seven columns of emotions

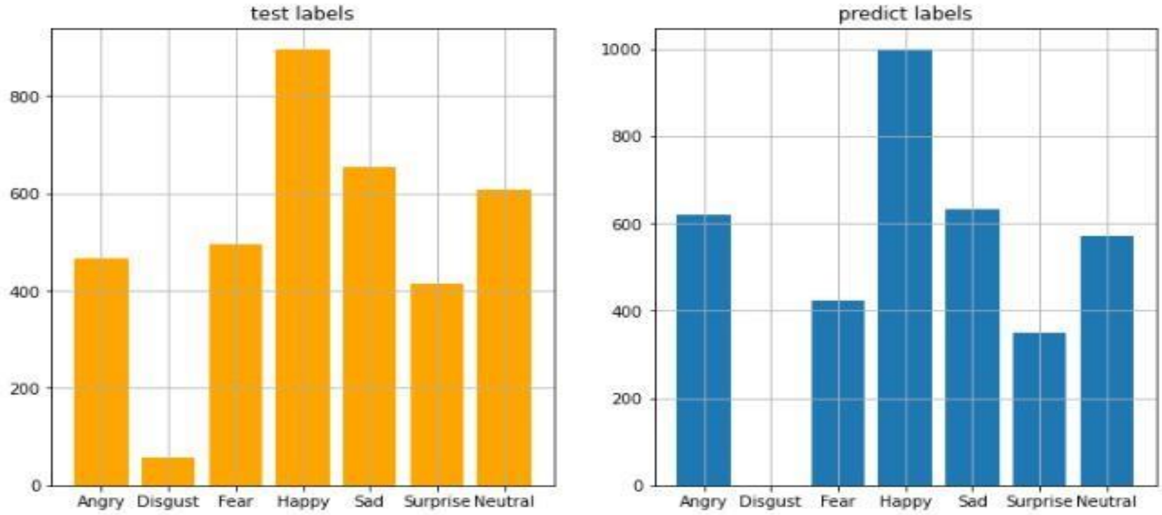
After prepare the dataset, we split our data to training, validation and test data.



```
Training       28709
PrivateTest     3589
PublicTest      3589
```

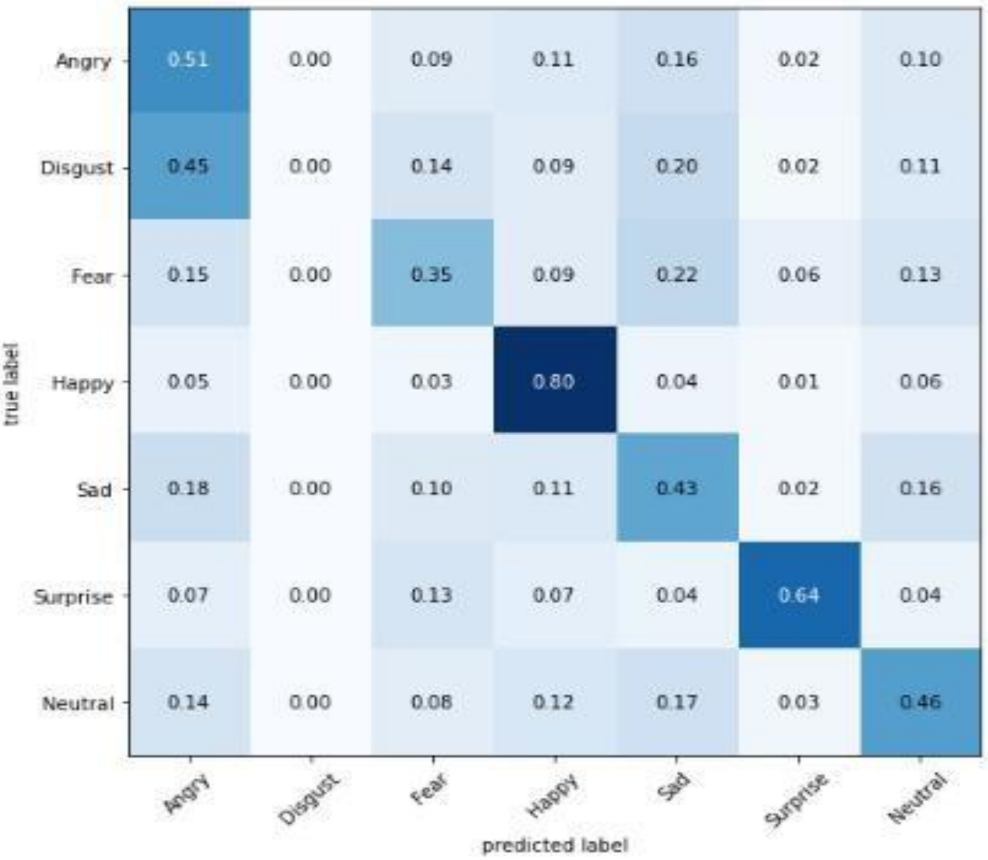
The next step is reshape and scale the images to 48 x 48 pixel grayscale image of faces. The faces have been automatically registered so that the face is more or less cantered and occupies about the same amount of space in each image. After scaling the images, we define the convolution neural network model with convolution layer, pooling layer, fully connected layer and using Rectified Linear Unit(ReLU) activation function to increase nonlinearity in the model.

After that, testing the model using test data and obtain predicted value.



Testing data with predicted value graphic

The accuracy score is about 54% on the test set and to improve the model in the future is to use a confusion matrix to extract the details.



Confusion matrix obtained from test data and predicted value

Output images

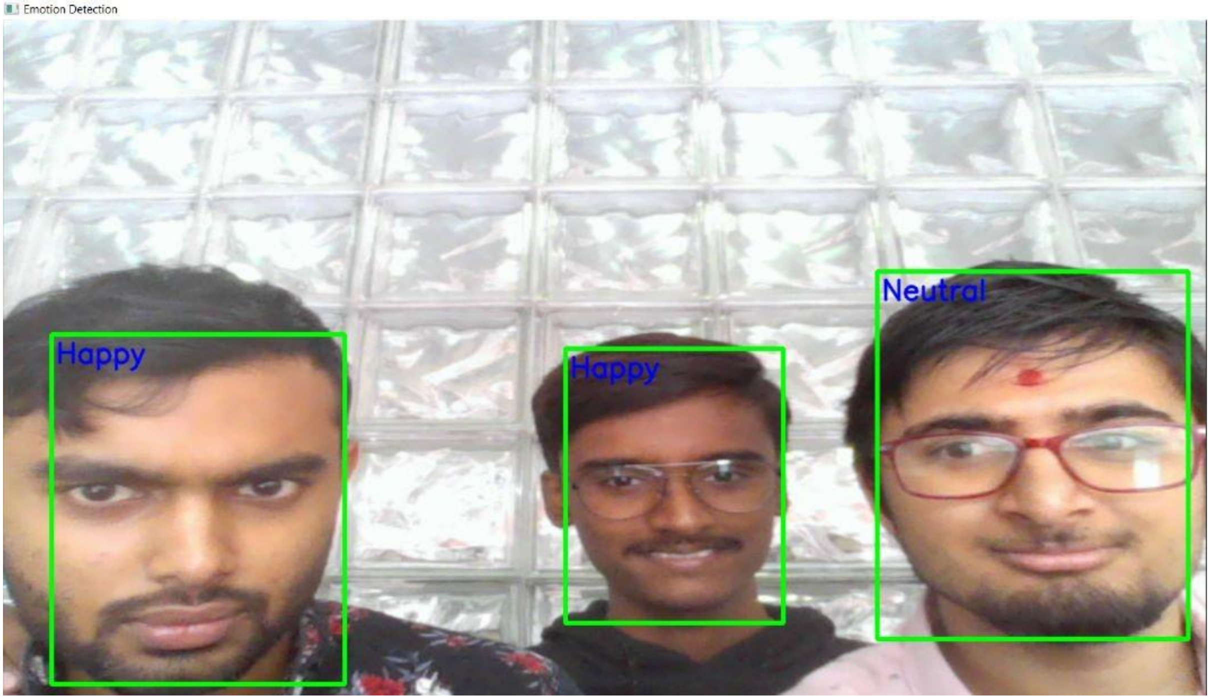


Fig 1

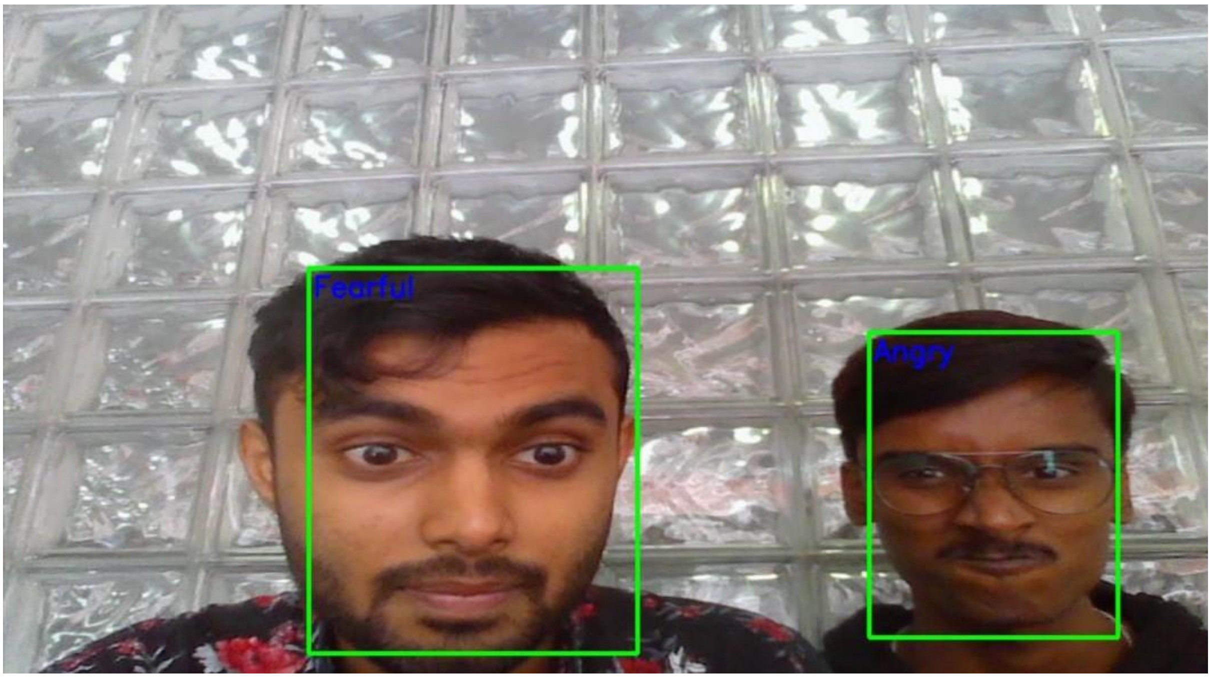


Fig 2

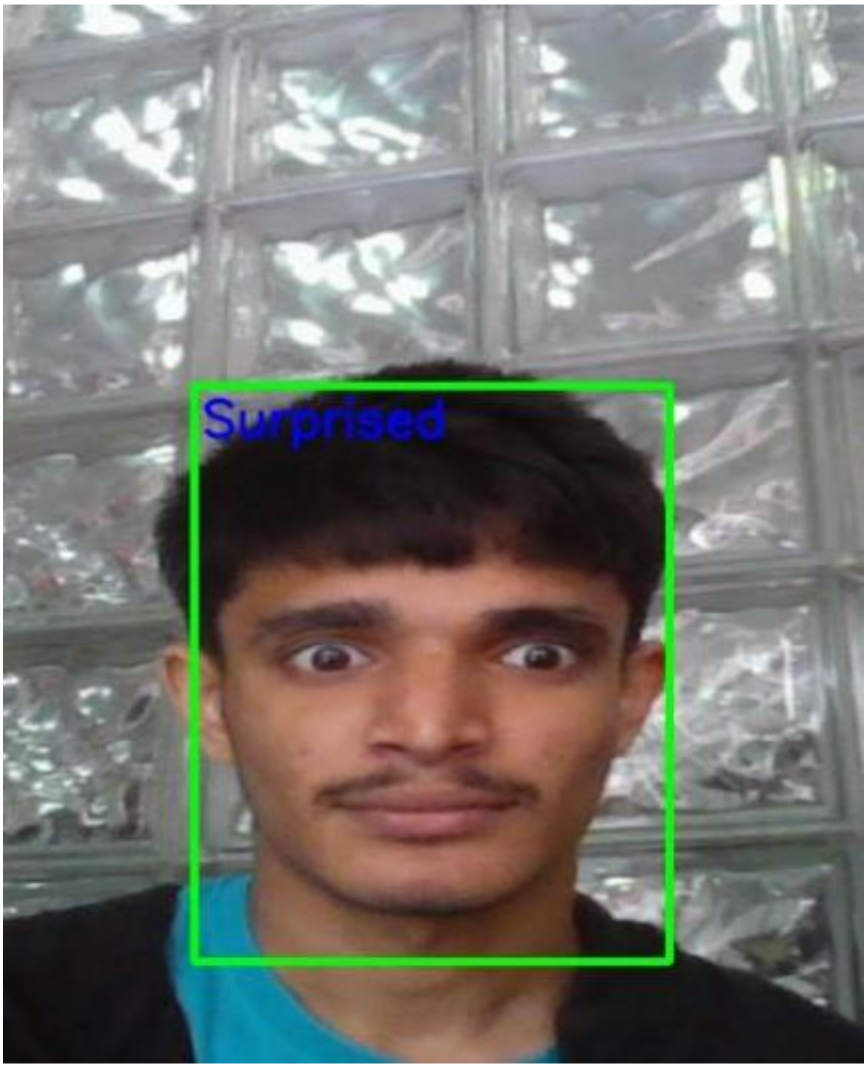


Fig 3

Project schedule

Sl No	Process	Date
1	Topic Selection	20-10-2022
2	Idea Presentation	03-11-2022
3	Code Implementation	01-12-2022
4	Report Making	10-12-2022
5	Report Submission	12-12-2022

Conclusion

Emotions are part of the result of physiological responses where the delivered signals will give different emotions. To recognize human emotion, an intelligent system must detect the face from the input images and extract the Haar Cascade features of the face using the Viola-Jones algorithm, then perform feature extraction, recognition process, and determine the emotion. The process of determining emotion that is used in this paper is the Convolutional Neural Network(CNN) model which has dropout layers as well as pooling to prevent overfitting of the model as well as increase efficiency. We are using the Facial Expressions Recognition-2013(FER-2013) dataset to test our model for determining the emotion that contains 35.887 data divided into seven emotions. The model that has been created gives 54% accuracy from recognizing seven human emotions.

Although the result accuracy is not high but can be used quite well to recognizing human emotions.

References

- 2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS)
- Recognizing human emotion using computer vision
- Jonathan., Andreas Pangestu Lim., Paoline., Gede Putra Kusuma., and Amalia Zahra, "Facial Emotion Recognition Using ComputerVision", Researchgate, September 2018.
- The dataset the used is Facial Expressions Recognition-2013(FER-2013)
- https://www.kaggle.com/msambare/fer2013
- https://github.com/datamagic2020
- https://www.mygreatlearning.com/blog/viola-jones-algorithm