Emotion Detection

##### MINI PROJECT REPORT

###### ***Submitted by***

##### AUMRUDH LAL KUMAR TJ (201806007)

**SANJAI KUMAAR M** **(201806040)**

*in*

# MACHINE LEARNING USING TENSORFLOW

**DEPARTMENT OF COMPUTER SCIENCE**

MEPCO SCHLENK ENGINEERING COLLEGE

SIVAKASI

##### DECEMBER 2021

Contents

[MACHINE LEARNING USING TENSORFLOW 1](#_Toc90064136)

[Introduction 3](#_Toc90064137)

[Motivation 3](#_Toc90064138)

[Project Description 4](#_Toc90064139)

[1. Convolutional Layer 4](#_Toc90064140)

[2. Pooling Layer 5](#_Toc90064141)

[3. Fully Connected Layer 5](#_Toc90064142)

[4. Dropout 5](#_Toc90064143)

[5. Activation Functions 5](#_Toc90064144)

[Code 6](#_Toc90064145)

[Output 8](#_Toc90064146)

[Conclusion 10](#_Toc90064147)

## Introduction

A widely accepted theory of basic emotions and their expressions, developed Paul Ekman, suggests we have six basic emotions. They include sadness, happiness, fear, anger, surprise and disgust.

Sadness is an emotional state characterized by feelings of disappointment, grief or hopelessness. Happiness is a pleasant emotional state that elicits feelings of joy, contentment and satisfaction. Fear a primal emotion that is important to survival and triggers a fight or flight response. Anger an emotional state leading to feelings of hostility and frustration**.** Surprise a brief emotional state, either positive or negative, following something unexpected. Disgust a strong emotion that results in feeling repulsed

## Motivation

If someone showed you a picture of a person and asked you to guess what they’re feeling, chances are you’d have a pretty good idea about it. What if your computer could do the same? It will be great if computer gives the output since it could be used for the large people with different emotions. This can be done using the Convolutional neural network.  Emotion recognition is an AI technology used to analyze a person's face and interpret their expressions. The emotion can be captured either from face or from verbal communication. In this work we focus on identifying human emotion from facial expressions

## Project Description

CNNs are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analyzing visual images. Their applications range from image and video recognition, image classification, medical image analysis, computer vision and natural language processing. There are two main parts to a CNN architecture

* A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction
* A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

## 



### 1. Convolutional Layer

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

### 2. Pooling Layer

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of Pooling operations. In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer

### 3. Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture. In this, the input image from the previous layers are flattened and fed to the FC layer. In this stage, the classification process begins to take place.

### 4. Dropout

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

### 5. Activation Functions

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network. It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, SoftMax, tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and SoftMax functions are preferred a for a multi-class classification, generally SoftMax is used.

## Code

import numpy as np

import cv2

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout, Flatten

from tensorflow.keras.layers import Conv2D

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.preprocessing.image import ImageDataGenerator

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"

)

validation\_generator = val\_datagen.flow\_from\_directory(

val\_dir,

target\_size=(48,48),

batch\_size=64,

color\_mode="grayscale",

class\_mode="categorical")

model = Sequential()

model.add(Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(48,48,1)))

model.add(Conv2D(64, kernel\_size=(3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.25))

model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(1024, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(7, activation='softmax'))

model.compile(loss='categorical\_crossentropy',optimizer=Adam(lr=0.0001, decay=1e-6),metrics=['accuracy'])

model\_info = model.fit\_generator(

train\_generator,

steps\_per\_epoch=28709 // 64,

epochs=50,

validation\_data=validation\_generator,

validation\_steps=7178 // 64)

model.save\_weights('model.h5')

cv2.ocl.setUseOpenCL(False)

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

cap = cv2.VideoCapture(0)

while True:

ret, frame = cap.read()

if not ret:

break

#bounding\_box = cv2.CascadeClassifier('/home/shivam/.local/lib/python3.6/site-packages/cv2/data/haarcascade\_frontalface\_default.xml')

bounding\_box = cv2.CascadeClassifier('C:/Users/imaum/AppData/Local/Programs/Python/Python37/Lib/site-packages/cv2/data/haarcascade\_frontalface\_default.xml')

gray\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

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 = 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))

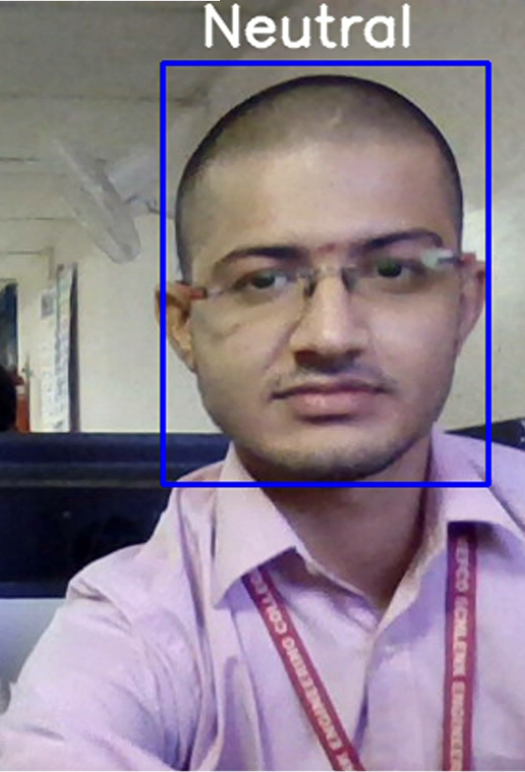
if cv2.waitKey(13) & 0xFF == ord('q'):

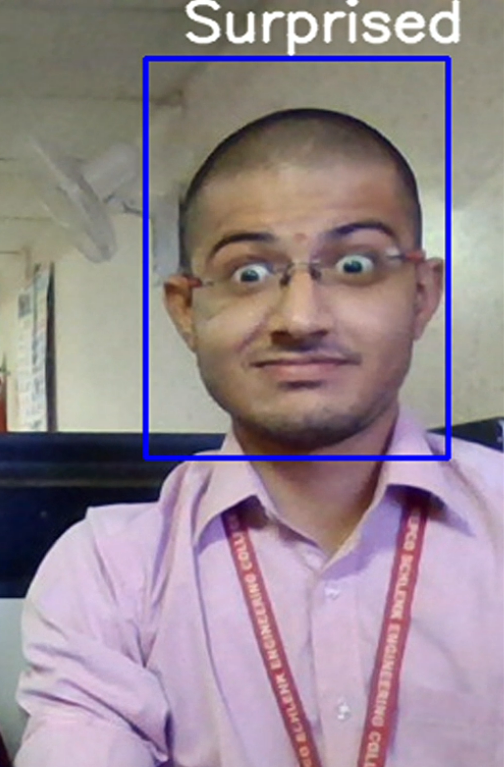
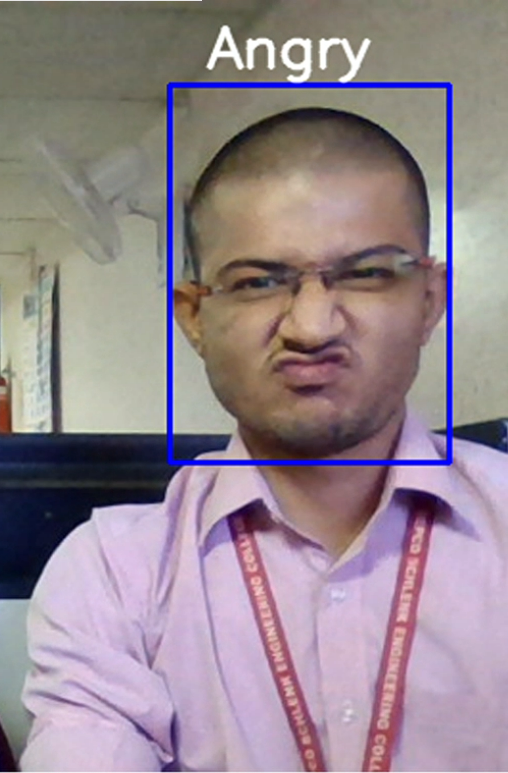
break

cap.release()

cv2.destroyAllWindows()

## Output



## Conclusion

Human emotion recognition plays an important role in the interpersonal relationship. ... Emotions are reflected from speech, hand and gestures of the body and through facial expressions. Hence extracting and understanding of emotion has a high importance of the interaction between human and machine communication. If we need to get to know the expression of large set of people then we could use deep learning techniques and detect their emotion.