GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Facial Emotion Recognition in Python

A Report for the Evaluation 3 of Project 2

Submitted by

NIKHIL KUMAR SINGH

(1613101447 / 16SCSE101202)

in partial fulfillment for the award of the degree

of

Bachelor of Technology

IN

Computer Science and Engineering

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Under the Supervision of

Mr. V. Gokul Rajan

Assistant Professor

APRIL/MAY-2020

TABLE OF CONTENTS

1. ABSTRACT

Humans share a universal and fundamental set of emotions which are exhibited through consistent facial expressions. An algorithm that performs detection, extraction, and evaluation of these facial expressions will allow for automatic recognition of human emotion in images and videos. This application is based on image processing and machine learning. In this application captured image from webcam is compared with the trained dataset available and then emotional state of the image will be displayed. Facial Emotion Recognition application is implemented using Convolution Neural Network (CNN). Dataset with seven facial expression labels as happy, sad, surprise, fear, anger, disgust, and neutral is used in this project. The application has achieved 56.77 % accuracy and 0.57 precision on testing dataset.

2. INTRODUCTION

A Facial expression is the visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person and plays a communicative role in interpersonal relations. 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. On a day to day basics humans commonly recognize emotions by characteristic features, displayed as a part of a facial expression. For instance happiness is undeniably associated with a smile or an upward movement of the corners of the lips.

(i) Overall Description: Generally human beings can convey intentions and emotions through nonverbal ways such as gestures, facial expressions and involuntary languages. This system can be significantly useful, nonverbal way for people to communicate with each other. The important thing is how fluently the system detects or extracts the facial expression from image.

(ii) Purpose: The system classifies facial expression of the same person into the basic emotions namely anger, disgust, fear, happiness, sadness and surprise. The main purpose of this system is efficient interaction between human beings and machines using eye gaze, facial expressions, cognitive modeling etc. Here, detection and classification of facial expressions can be used as a natural way for the interaction between man and machine. And the system intensity vary from person to person and also varies along with age, gender, size and shape of face, and further, even the expressions of the same person do not remain constant with time.

(iii) Motivations and Scope: The universality of these expressions means that facial emotion recognition is a task that can also be accomplished by computers. Furthermore, like many other important tasks, computers can provide advantages over humans in analysis and problem-solving. Computers that can recognize facial expressions can find application where efficiency and automation can be useful, including in entertainment, social media, content analysis, criminal justice, and healthcare. For example, content providers can determine the reactions of a consumer and adjust their future offerings accordingly.

3.LITERATURE REVIEW

S.No.	Name	Proposed Model	Description	Year
1.)	Ekman	Set of Emotions	Defined six rudimentary Emotions: Joy, Sadness, anger, fear, disgust and surprise	1970
2.)	Y. Fan, X. Lu, D. Li, and Y. Liu.	Video-based Emotion Recognition Using CNN-RNN and C3D Hybrid	Achieved accuracy 59.02% (without using any additional Emotion labeled	2016

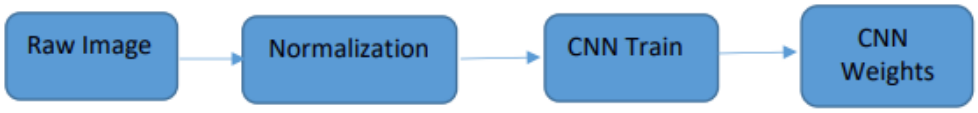
		Networks	video clips in training set) which is the best till now.	
3.)	Zixing Zhang, Fabien Ringeval, Fabien Ringeval, Eduardo Coutinho, Erik Marchi and Björn Schüller	Semi-Supervised Learning (SSL) technique	Delivers a strong performance in the classification of high/low emotional arousal (UAR = 76.5%), and significantly outperforms traditional SSL methods by at least 5.0% (absolute gain).	2016
4.)	Deepika Ishwar	Region Based Segmentation	Face-Feature Extraction Method (Using Eye and Mouth) Using Bezier	2015

			Curve	

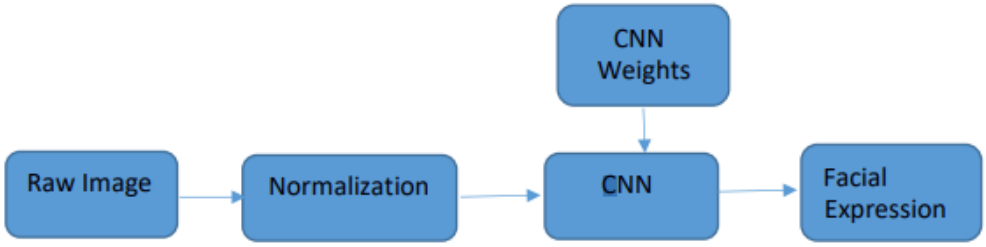
4. PROPOSED SYSTEM

The facial expression recognition system is implemented using convolutional neural network.

The block diagram of the system is shown in following figures:



(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, an 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.

5. IMPLEMENTATION

To get the application run python file real_time_video.py . It will run with pre-trained model.

Loading and Processing

Load_and_process.py

```
import pandas as pd
import cv2
import numpy as np


dataset_path = 'fer2013/fer2013/fer2013.csv'
image_size=(48,48)

def load_fer2013():
        data = pd.read_csv(dataset_path)
        pixels = data['pixels'].tolist()
        width, height = 48, 48
        faces = []
        for pixel_sequence in pixels:
            face = [int(pixel) for pixel in pixel_sequence.split(' ')]
            face = np.asarray(face).reshape(width, height)
            face = cv2.resize(face.astype('uint8'),image_size)
            faces.append(face.astype('float32'))
```

```
        faces = np.asarray(faces)
        faces = np.expand_dims(faces, -1)
        emotions = pd.get_dummies(data['emotion']).as_matrix()
        return faces, emotions

def preprocess_input(x, v2=True):
    x = x.astype('float32')
    x = x / 255.0
    if v2:
        x = x - 0.5
        x = x * 2.0
    return x
```

train_emotion_classifier.py

```
"""
Description: Train emotion classification model
"""

from keras.callbacks import CSVLogger, ModelCheckpoint, EarlyStopping
from keras.callbacks import ReduceLROnPlateau
from keras.preprocessing.image import ImageDataGenerator
from load_and_process import load_fer2013
from load_and_process import preprocess_input
from models.cnn import mini_XCEPTION
from sklearn.model_selection import train_test_split

# parameters
batch_size = 32
```

```
num_epochs = 10000
input_shape = (48, 48, 1)
validation_split = .2
verbose = 1
num_classes = 7
patience = 50
base_path = 'models/'

# data generator
data_generator = ImageDataGenerator(
                    featurewise_center=False,
                    featurewise_std_normalization=False,
                    rotation_range=10,
                    width_shift_range=0.1,
                    height_shift_range=0.1,
                    zoom_range=.1,
                    horizontal_flip=True)

# model parameters/compilation
model = mini_XCEPTION(input_shape, num_classes)
model.compile(optimizer='adam', loss='categorical_crossentropy',
            metrics=['accuracy'])
model.summary()
```

```
    # callbacks
log_file_path = base_path + '_emotion_training.log'
csv_logger = CSVLogger(log_file_path, append=False)
early_stop = EarlyStopping('val_loss', patience=patience)
reduce_lr = ReduceLROnPlateau('val_loss', factor=0.1,
                              patience=int(patience/4), verbose=1)
trained_models_path = base_path + '_mini_XCEPTION'
model_names = trained_models_path + '.{epoch:02d}-{val_acc:.2f}.hdf5'
model_checkpoint = ModelCheckpoint(model_names, 'val_loss', verbose=1,
                                   save_best_only=True)
callbacks = [model_checkpoint, csv_logger, early_stop, reduce_lr]

# loading dataset
faces, emotions = load_fer2013()
faces = preprocess_input(faces)
num_samples, num_classes = emotions.shape
xtrain, xtest,ytrain,ytest = train_test_split(faces, emotions,test_size=0.2,shuffle=True)
model.fit_generator(data_generator.flow(xtrain, ytrain,
                                        batch_size),
                    steps_per_epoch=len(xtrain) / batch_size,
                    epochs=num_epochs, verbose=1, callbacks=callbacks,
                    validation_data=(xtest,ytest))
```

real_time_video.py

```
from keras.preprocessing.image import img_to_array
import imutils
import cv2
```

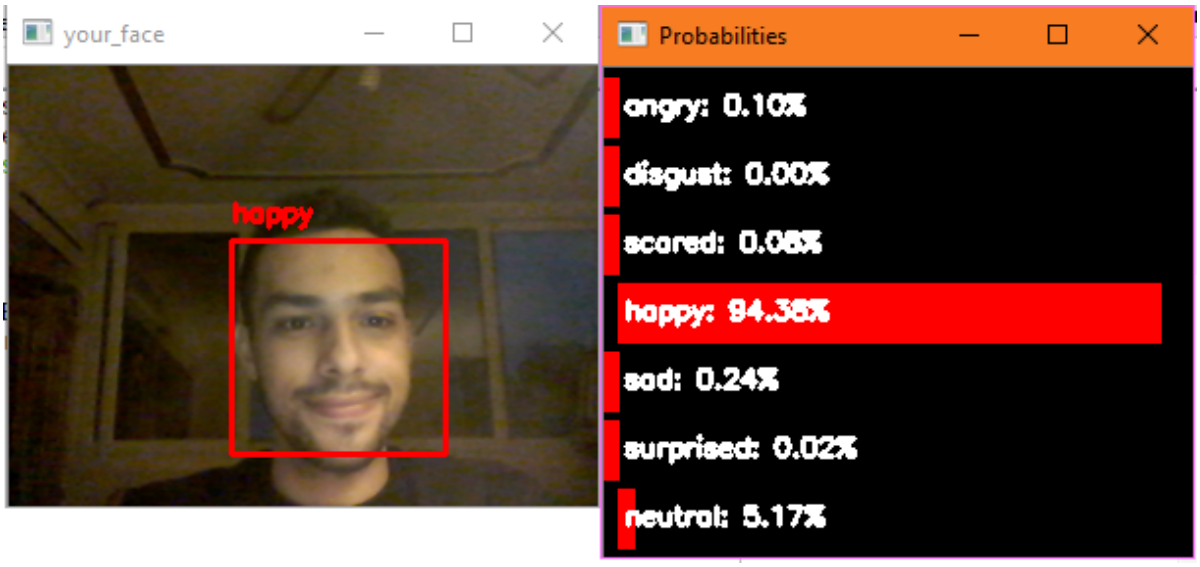
```
from keras.models import load_model
import numpy as np

# parameters for loading data and images
detection_model_path = 'haarcascade_files/haarcascade_frontalface_default.xml'
emotion_model_path = 'models/_mini_XCEPTION.102-0.66.hdf5'

# hyper-parameters for bounding boxes shape
# loading models
face_detection = cv2.CascadeClassifier(detection_model_path)
emotion_classifier = load_model(emotion_model_path, compile=False)
EMOTIONS = ["angry" ,"disgust","scared", "happy", "sad", "surprised",
 "neutral"]



#feelings_faces = []
#for index, emotion in enumerate(EMOTIONS):
  # feelings_faces.append(cv2.imread('emojis/' + emotion + '.png', -1))

# starting video streaming
cv2.namedWindow('your_face')
camera = cv2.VideoCapture(0)
while True:
   frame = camera.read()[1]
   #reading the frame
   frame = imutils.resize(frame,width=300)
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
   faces =
face_detection.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),
```

```
flags=cv2.CASCADE_SCALE_IMAGE)

    canvas = np.zeros((250, 300, 3), dtype="uint8")
    frameClone = frame.copy()
    if len(faces) > 0:
        faces = sorted(faces, reverse=True,
        key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
        (fX, fY, fW, fH) = faces
                # Extract the ROI of the face from the grayscale image, resize it to a fixed
28x28 pixels, and then prepare
            # the ROI for classification via the CNN
        roi = gray[fY:fY + fH, fX:fX + fW]
        roi = cv2.resize(roi, (64, 64))
        roi = roi.astype("float") / 255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi, axis=0)



        preds = emotion_classifier.predict(roi)[0]
        emotion_probability = np.max(preds)
        label = EMOTIONS[preds.argmax()]
    else: continue



    for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):
            # construct the label text
            text = "{}: {:.2f}%".format(emotion, prob * 100)


            # draw the label + probability bar on the canvas
```

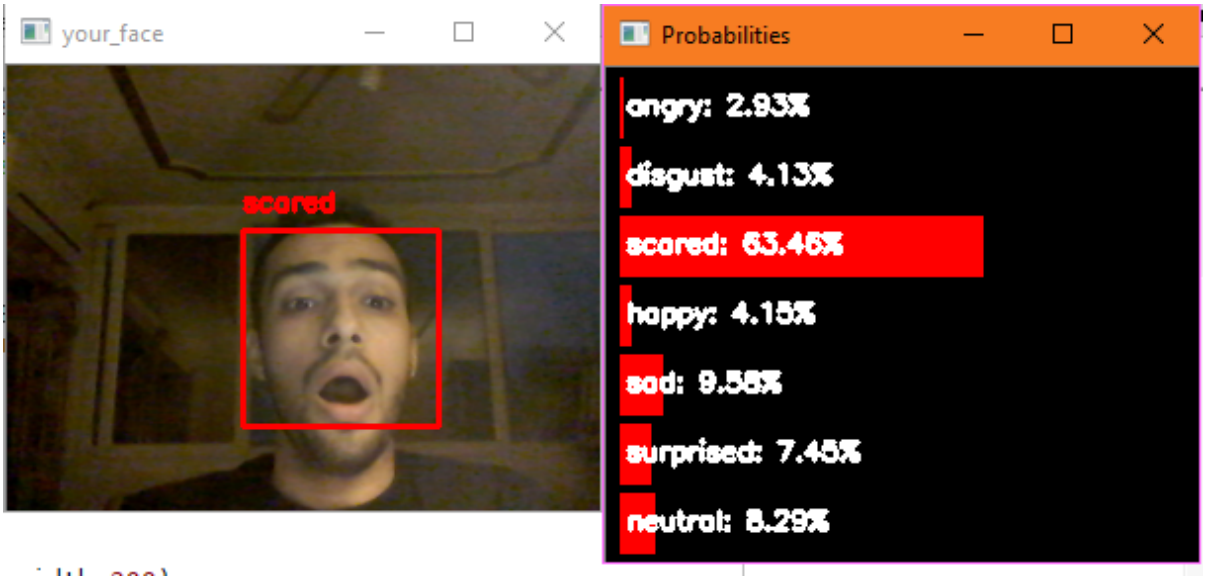
```
            # emoji_face = feelings_faces[np.argmax(preds)]


            w = int(prob * 300)
            cv2.rectangle(canvas, (7, (i * 35) + 5),
            (w, (i * 35) + 35), (0, 0, 255), -1)
            cv2.putText(canvas, text, (10, (i * 35) + 23),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45,
            (255, 255, 255), 2)
            cv2.putText(frameClone, label, (fX, fY - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
            cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY + fH),
                    (0, 0, 255), 2)
#    for c in range(0, 3):
#        frame[200:320, 10:130, c] = emoji_face[:, :, c] * \
#        (emoji_face[:, :, 3] / 255.0) + frame[200:320,
#        10:130, c] * (1.0 - emoji_face[:, :, 3] / 255.0)


    cv2.imshow('your_face', frameClone)
    cv2.imshow("Probabilities", canvas)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

camera.release()
cv2.destroyAllWindows()
```

6. OUTPUT

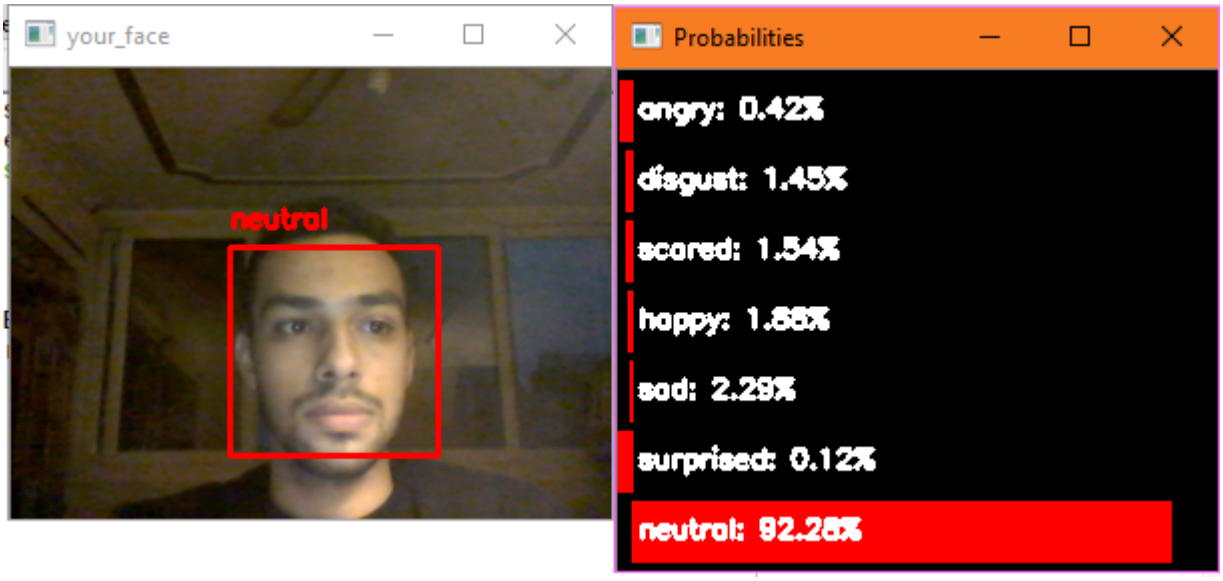
HAPPY FACE



SCARED FACE



NEUTRAL FACE



7. CONCLUSION

Feeling acknowledgment of human face is the most widely recognized procedure in biometric applications. In this undertaking, we investigated a novel method for grouping human feelings from outward appearances. In this way a neural network based

arrangement joined with image processing was proposed to order the six all inclusive feelings: Happy, Sad, Anger, Disgust, neutral and scared. The fundamental goal is to build up a programmed feeling identification framework with the powerful ongoing face detection algorithm and segmentation process. Proposed framework is tried on video outlines from the single video for facial feelings and palatable outcomes are gotten.

8. REFERENCES

[1] Pham, T., and Worring, M., Face detection methods: A critical evaluation. ISIS Technical Report Series, University of Amsterdam, 11, 2000.

[2] Yang, M. H., Ahuja, N., and Kriegman, D., A survey on face detection methods, 1999

[3] Cohen, I., Garg, A., and Huang, T. S., Emotion from facial expressions using multilevel hmm. In Neural Information Processing Systems, 2000.

[4] Gu, H., Su, G., and Du, C., Feature points extraction from faces. 2003. 11) Lisetti, C. L., and Rumelhart, D. E., Facial expression recognition using a neural network. In Proceedings of the Eleveth International FLAIRS Conference. Menlo Park, pages 328–332. AAAI Press, 1998.

[5] Samir K. Bandyopadhyay proposed, "A Method For Face Segmentation, Facial Feature Extraction And Tracking" IJCSET | April 2011 | vol.no:1, issue 3,137-139

[6] Prof. D. K Shah, Vishal B. Mokashi, "Face Detection and Recognition are Using Recorded Videos", International Journal of Emerging Technology and Advanced Engineering website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 certified journal, Volume 5, Issue 2, February 2015)

[7] Moon Hwan Kim, Young Hoon Joo and Jin BaePark, “Emotion Detection

Algorithm Using Frontal Face Image", IEEE 2013

[8] Leh Luoh, Chih-Chang Huang and Hsueh-YenLiu :Image Processing based emotion recognition", IEEE 6,December 2011.