ACTIVENESS MEASURE USING FACE RECOGNITION BASED ON EMOTIONS

A PROJECT REPORT

Submitted by

SHIVAYAVASHILAXMIPRIYA.S
(Reg. No: 20S030)

of

5 Year Integrated M.Sc., (Data Science)

in

DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

THIAGARAJAR COLLEGE OF ENGINEERING

(A Govt. Aided Autonomous Institution affiliated to Anna University)

MADURAI – 625015

SEPTEMBER 2022

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI

DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

BONAFIDE CERTIFICATE

Certified that this project report "ACTIVENESS MEASURE USING FACE RECOGNITION BASED ON EMOTIONS" is the bonafide work of SHIVAYAVASHILAXMIPRIYA.S (20S030) , Fifth Semester student of 5 Year Integrated MSc (Data Science) Degree Programme, who carried out the project under my supervision from 17.07.2022 to 22.07.2022 during the academic year 2022-2023.

The project report was submitted to the department on 22/09/2022 for evaluation/assessment.

<sign>

Dr. S. Parthasarathy
Professor & Head
Department of Applied Mathematics and Computational Science

<sign>

Prof.P.Sharmila
Project Guide
Assistant Professor in Data Science
Department of Applied Mathematics and Computational Science

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.No.	Abbreviation /Acronym	Description
1.	CNN	Convolutional Neural networks

Page No

1.Introduction:

The project aims to find the activeness of the students based on their emotions expressed by their faces. Students need to deal with a lot of tasks every day. Most of the students in their college life undergo stress and depression. At least one student commits suicide every hour in India. India lost more than 1.7 lakhs of students in last 25 years.

The face expressions are natural and direct means for human beings to communicate their emotions and intentions. This is where Deep learning provides a solution to find their emotions through their faces which can be used to provide them with proper guidance and assistance. Facial expression recognition is the task of classifying the expressions shown by faces into various categories such as happy, sad, neutral, disgust, fear, surprise and angry. These categories can be grouped into active and non-active types of students.

Deep Learning- Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modelling.

The Convolutional neural networks are one of the Deep Learning methodologies which is used to classify the person as active and non-active. CNN is a type of artificial neural network, which is widely used for image recognition and classification. It plays a vital role in diverse computer vision tasks.

The facial expressions are analyzed and classified using 2 methodologies:
1.Using Deep Face framework
2.Building a 4-layer sequential CNN model
The Deep Face framework classified the images more accurately when compared to the sequential model.

Deep Face framework- Deepface is a lightweight face recognition and facial attribute analysis framework for python. It is a hybrid framework wrapping various state of models like VGG-Face, OpenFace, Google FaceNet, Facebook DeepFace, Dlib, ArcFace and SFace.

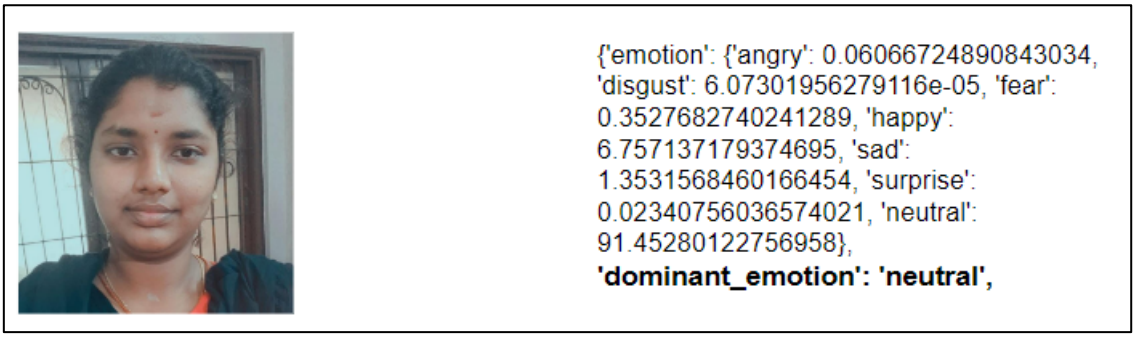

Fig 1.1: DeepFace model

Sequential CNN model- Sequential is the easiest way to build a model in Keras. It allows you to build a model layer by layer. The 'add()' function is used to add layers to the model.

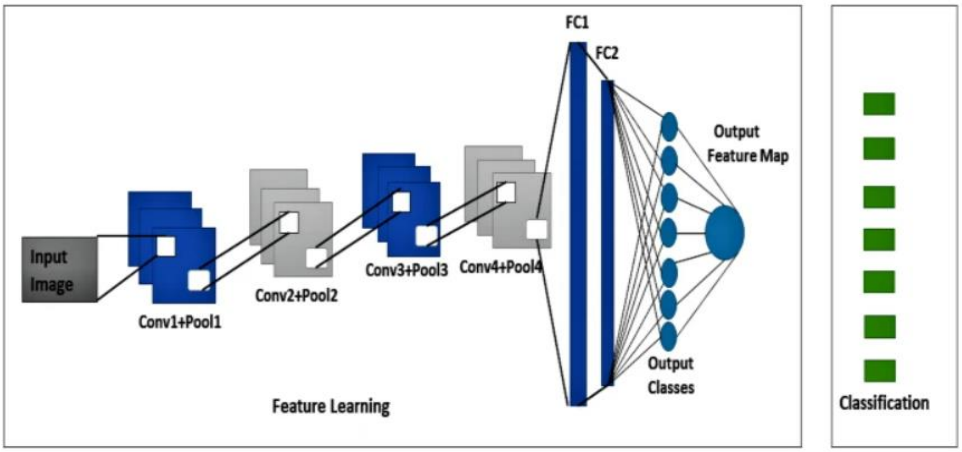

Fig 1.2: CNN model

2.Objective of the Project:

The main objective of the project is to classify the students as active and inactive based on the emotions shown by their faces. The students show different emotions based on their moods and feelings. Capturing these emotions and analyzing them will help the staffs to care more for the students who are constantly inactive and provide them with proper guidance.

This will help the students to excel in their carrier by providing them with clear guidance and assistance.
This activeness detection system can be used in classes or during attendance sessions to scan the faces and find their activeness. This will help the staffs to identify the students with dull and inactive moods and guide them properly.

The emotions shown by common human can be classified into 7 i.e. Happy, Sad, Disgust, Fear, Neutral, Angry and Surprise. These emotions Happy, Neutral and Surprise are classified as active and the remaining as inactive.

Then the status of activeness is stored in the excel on a daily basis. This can help the teachers to analyze the students who are constantly in inactive status and help them with proper guidance.

3.Description of the Project:

Programming language used: Python 3.9.6
Programming platforms used: Visual Studio Code, Kaggle Kernel
Other software: Excel for storing status of activeness

To find the emotions of the faces two models were developed and the one with more accuracy was used to classify.

i) DeepFace library:

Deepface is a facial recognition and attributes analysis framework for python created by the artificial intelligence research group at Facebook in 2015. Keras and TensorFlow inspire this library's core components. It is a hybrid face recognition framework that uses state-of-the-art models for analysis such as VGG-Face, Google FaceNet, Facebook Deepface, all wrapped together in one.

Through this library, the face detection, face recognition and the emotion detection can be done in images as well as in a real time video.

The common steps include
->Installing DeepFace library
->Importing images
->Face detection
->Face recognition
->Emotion recognition
->Other Facial features analysis
->Find dominant emotion

ii) Convolutional Neural Networks:

A great way to use deep learning to classify images is to build a convolutional neural network (CNN). The Keras library in Python makes it pretty simple to build a CNN.

The preprocessing required in a CNN is much lower than other classification algorithms. The architecture of a CNN is analogous to that of the connectivity pattern of neurons in the human brain and was inspired by the organization of the visual cortex. One role of a CNN is to reduce images into a form which is easier to process without losing features that are critical for good prediction.

The various CNN operations that are generally performed with the data are:

1. Convolution operation:
   The convolution operation is used to extract the high level features such as edges from the given input image.
2. Pooling operation:
   Pooling is a down-sampling operation that reduces the dimensionality of the feature map. The rectified feature map now goes through a pooling layer to generate a pooled feature map.

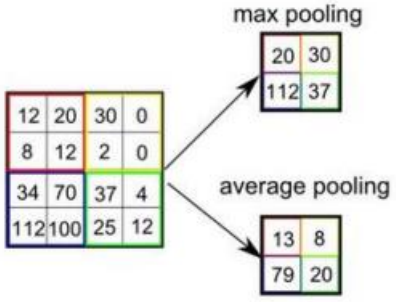

Fig 3.1: Pooling

3. Relu operation:

   ReLU stands for the rectified linear unit. Once the feature maps are extracted, the next step is to move them to a ReLU layer. ReLU performs an element-wise operation and sets all the negative pixels to 0. It introduces non-linearity to the network, and the generated output is a rectified feature map.

4. SoftMax:

   Softmax-the function at last layer of neural network which calculates the probabilities distribution of the event over ‘n’ different events.

5. Dropout operation:

   Dropout is used to avoid overfitting. Overfitting in an ML model happens when the training accuracy is much greater than the testing accuracy. Dropout refers to ignoring neurons during training so they are not considered during a particular forward or backward pass leaving a reduced network.

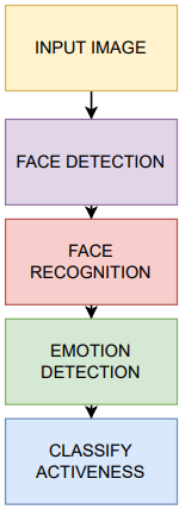

Fig 3.2: Model flowchart

4.Implementation of the project:

1. DeepFace model:

1.Loading the necessary libraries and capturing the faces through webcam:

```
import numpy as np
from datetime import datetime
import face_recognition as fr
from deepface import DeepFace
import cv2
import csv
import pytz
import time

now = datetime.now()

current_date = now.strftime("%Y-%m-%d")


video_capture=cv2.VideoCapture(0)
```

Fig 4.1: Importing packages

2.Loading the images and encoding them for face recognition and open a csv file for recording the status of the recognized face.

```
shivaya_image=fr.load_image_file("photos\shivaya.jpeg")
shivaya_encoding=fr.face_encodings(shivaya_image)[0]

sharmila_image=fr.load_image_file("photos\sharmila.jpeg")
sharmila_encoding=fr.face_encodings(sharmila_image)[0]

shanmathi_image=fr.load_image_file("photos\Shanmathi.jpeg")
shanmathi_encoding=fr.face_encodings(shanmathi_image)[0]

dona_image=fr.load_image_file("photos\Dona.jpeg")
dona_encoding=fr.face_encodings(dona_image)[0]

harinee_image=fr.load_image_file("photos\Harinee.jpeg")
harinee_encoding=fr.face_encodings(harinee_image)[0]

known_face_encodings=[shivaya_encoding,sharmila_encoding,shanmathi_encoding,dona_encoding,harinee_encoding]
known_face_names=["Shivaya","Sharmila","Shanmathi","Dona","Harinee"]
face_names = []
students = known_face_names.copy()
now = datetime.now()
current_date = now.strftime("%Y-%m-%d")

f = open(current_date + '.csv', 'w+',encoding='UTF8', newline='')
lnwriter = csv.writer(f)
```

Fig 4.2: Loading images and encoding faces

3.Once the face is detected and recognized using face recognition library the emotions are recognized using DeepFace library and the activeness is classified.

```
while True:
    times=now.strftime('%I:%M:%S')
    ret,frame=video_capture.read()
    rgb_frame=frame[:,:,::-1]
    face_locations=fr.face_locations(rgb_frame)
    face_encodings=fr.face_encodings(rgb_frame,face_locations)
    result = DeepFace.analyze(frame, actions=['emotion'],enforce_detection=False)
    font=cv2.FONT_HERSHEY_SIMPLEX
    for (top, right, bottom, left), face_encoding in zip(face_locations,face_encodings):
        matches=fr.compare_faces(known_face_encodings,face_encoding)

        name="Unknown"

        face_distances=fr.face_distance(known_face_encodings,face_encoding)

        best_match_index=np.argmin(face_distances)

        if matches[best_match_index]:
            name=known_face_names[best_match_index]
        if result['dominant_emotion'] in ['happy','surprise','neutral']:
            status='active'
        elif result['dominant_emotion'] in ['fear','disgust','angry','sad']:
            status='Not active'

        face_names.append(name)
        if name in known_face_names:
            if name in students:
                students.remove(name)
                lnwriter.writerow([name,times,current_date,status])

        cv2.rectangle(frame,(left,top),(right,bottom),(0,0,255),2)

        cv2.rectangle(frame,(left,bottom -35),(right,bottom),(0,0,255),cv2.FILLED)

        cv2.putText(frame,name,(left +6,bottom -6),font,1.0,(255,255,255),1)
```

Fig 4.3: Face recognition and Emotion detection of captured images

4.Output images:

The faces are recognized and the frames were displayed. The figure shows the sample output of 2 recognized images and the status of all the classified images are updates in the created csv file along with the name, date and time.

Fig 4.4: Recognizing faces

1	Shivaya	07:20:56	18-09-2022	active
2	Harinee	07:20:56	18-09-2022	active
3	Sharmila	07:20:56	18-09-2022	active
4	Dona	07:20:56	18-09-2022	Not active
5	Shanmathi	07:20:56	18-09-2022	active

Fig 4.5: Storing results in excel

2. 4-layer sequential CNN model:

1.The dataset used here FER 2013 which is a publicly available dataset used for face emotion recognition. The required libraries and the dataset are loaded.

The epochs indicate the number of passes of the entire training dataset the machine learning algorithm has completed.

```
import sys, os
import pandas as pd
import numpy as np

from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D, BatchNormalization,AveragePooling2D
from keras.losses import categorical_crossentropy
from tensorflow.keras.optimizers import Adam
from keras.regularizers import l2
from keras.utils import np_utils
```

```
IMG_HEIGHT=48
IMG_WIDTH = 48
num_features = 64
num_labels = 7
batch_size = 64
epochs = 100

train_data_dir='../input/face-expression-recognition-dataset/images/train'
validation_data_dir='../input/face-expression-recognition-dataset/images/validation'
```

Fig 4.6: Importing libraries

2. The data is preprocessed and rescaled to build the CNN model

```
train_datagen = ImageDataGenerator(
rescale=1./255,
rotation_range=30,
shear_range=0.3,
zoom_range=0.3,
horizontal_flip=True,
fill_mode='nearest')
```

```
validation_datagen = ImageDataGenerator(rescale=1./255)
```

+ Code + Markdown

```
train_generator = train_datagen.flow_from_directory(
train_data_dir,
color_mode='grayscale',
target_size=(IMG_HEIGHT, IMG_WIDTH),
batch_size=batch_size,
class_mode='categorical',
shuffle=True)
```

Fig 4.7: Preprocessing

3.The sequential model is built with the preprocessing techniques of ReLU operation, Max pooling and the layer is at last flatten into an array if vectors and then the model is compiled with the Adam optimizer.

```
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(0.1))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.1))

model.add(Conv2D(256, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.1))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))

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

model.compile(optimizer = 'adam', loss='categorical_crossentropy', metrics=['accuracy'])
print(model.summary())
```

Fig 4.8: Model development

4.The number of epochs is set to 70 and the model is fitted with the training and validation data in the FER dataset.

```
train_path = "../input/face-expression-recognition-dataset/images/train"
test_path = "../input/face-expression-recognition-dataset/images/validation"

num_train_imgs = 0
for root, dirs, files in os.walk(train_path):
    num_train_imgs += len(files)

num_test_imgs = 0
for root, dirs, files in os.walk(test_path):
    num_test_imgs += len(files)


epochs=70

history=model.fit(train_generator,
                  steps_per_epoch=num_train_imgs//batch_size,
                  epochs=epochs,
                  validation_data=validation_generator,
                  validation_steps=num_test_imgs//batch_size)

model.save('emotion_detection_model_70epochs.h5')
```

```
Epoch 69/70
900/900 [==============================] - 39s 44ms/step - loss: 1.0370 - accuracy: 0.6058 - val_loss: 1.0223 - val_accuracy: 0.6264
Epoch 70/70
900/900 [==============================] - 40s 44ms/step - loss: 1.0443 - accuracy: 0.6066 - val_loss: 1.0276 - val_accuracy: 0.6239
```

Fig 4.9:Model fitting

5. The model is saved as a h5 file and it is used in the face emotion recognition model which captures the images using webcam and classifies them as active and inactive. The haarcascade classifier is used to detect the faces and the emotion is detected from the faces.

```
face_classifier =cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
emotion_model = load_model('emotion_detection_model_100epochs.h5')

class_labels=['Angry','Disgust', 'Fear', 'Happy','Neutral','Sad','Surprise']

cap=cv2.VideoCapture(0)

while True:
    ret,frame=cap.read()
    labels=[]

    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces=face_classifier.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray=gray[y:y+h,x:x+w]
        roi_gray=cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_AREA)

        #Get image ready for prediction
        roi=roi_gray.astype('float')/255.0  #Scale
        roi=img_to_array(roi)
        roi=np.expand_dims(roi,axis=0)  #Expand dims to get it ready for prediction (1, 48, 48, 1)

        preds=emotion_model.predict(roi)[0]  #Yields one hot encoded result for 7 classes
        label=class_labels[preds.argmax()]  #Find the label
        label_position=(x,y)

        if label in ['Happy','Surprise','Neutral']:
            status='Active'
        elif label in ['Fear','Disgust','Sad','Angry']:
            status='Not active'


        cv2.putText(frame,status,label_position,cv2.FONT_HERSHEY_SIMPLEX,1,(0,255,0),2)

    cv2.imshow('Emotion Detector', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Fig 4.10: Importing model and face classifying

6.The classification shown by the developed CNN model is shown in the figure

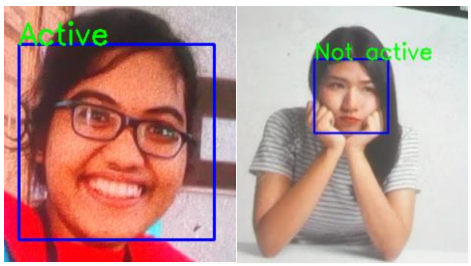

Fig 4.11: Output

Findings from the developed 2 models:

The DeepFace model shows an accuracy an accuracy of 97% and the 4-layer CNN model shows an accuracy of 60%.

5.Significance of the Project:

One of the main reasons that activation system is important is to monitor the students and provide them with proper guidance and care. This activation system is very useful as it predicts the activeness and notifies it earlier and helps the students to get proper guidance.
The main aim of the project is to reduce the suicides among the school and college students. As prevention is better than cure, this model detects the activeness and notifies the teacher to help or provide proper guidance to the respective students.

Challenges:

The emotion detection models developed could not classify the images clearly if there are no proper lights and brightness in the surroundings.

6.Conclusion:

In both the activeness detection systems built using 2 models, the common steps carried out were Face detection, recognition of the faces and then classifying the emotions shown on the faces. The DeepFace framework works better with 97% and here the faces were recognized using Face recognition library and then the emotion was classified. The CNN model was developed with the famous FER 2013 dataset which contains images showing 7 different emotions such as happy, sad, angry, neutral, surprise, fear and disgust. The 4-layer CNN model was developed and stored as a h5 file. Then the h5 model file was used to classify the faces using webcam. Here the haarcascade xml file is used to detect the faces and then the developed CNN model is used classify the status of activeness. This model attained an accuracy of 60%. Thus, the DeepFace model is the better among the two models.

The future works include building more complex CNN models with more epochs and layers to increase the accuracy of the model.

7.PROJECT WORKSHEET / DIARY

	Date	Topics learned / Activity carried out / Task completed / Online /E-resourcesaccessed
WEEK 1	10.08.2022	Topic selection from Smart India Hackathon and reporting to faculty guide.
	11.08.2022	
	12.08.2022	
	13.08.2022	Reading articles about face recognition
	14.08.2022	Reading articles about Emotion recognition
	15.08.2022	Reading articles about Emotion recognition

	Date	Topics learned / Activity carried out / Task completed / Online /E-resources accessed
WEEK 2	16.08.2022	Reading articles about Emotion recognition
	17.08.2022	Listing the methods and algorithms for emotion recognition
	18.08.2022	Installing software and packages
	19.08.2022	Searching for dataset
	20.08.2022	Learn basic face detection coding methods
	21.08.2022	Learn basic face recognition coding methods

	Date	Topics learned / Activity carried out / Task completed / Online /E-resourcesaccessed
WEEK 3	22.08.2022	Importing and working with face recognition library
	23.08.2022	Coding for face recognition and test it on video capture images
	24.08.2022	Internal Exams
	25.08.2022	
	26.08.2022	
	27.08.2022	

	Date	Topics learned / Activity carried out / Task completed / Online /E-resourcesaccessed
WEEK 4	28.08.2022	Build Deepface face recognition and emotion detection model
	29.08.2022	Testing the model on various images and faces
	30.08.2022	Testing the model on various images and faces
	31.08.2022	Reading the history and basics of CNN
	1.09.2022	Reading articles about CNN methods
	2.09.2022	Implementing a basic CNN model

	Date	Topics learned / Activity carried out / Task completed / Online /E-resourcesaccessed
WEEK 5	3.09.2022	Finding dataset for CNN model
	4.09.2022	1st Review-feedbacks and comments were noted
	5.09.2022	Using FER dataset and preprocessing Model Building
	6.09.2022	Importing the model for emotion recognition
	7.09.2022	Analyze the results
	8.09.2022	Increase the number of epochs and rerun the model

	Date	Topics learned / Activity carried out / Task completed / Online /E-resources accessed
WEEK 6	9.09.2022	Finding the accuracy and loss of the model
	10.09.2022	Comparing the DeepFace and CNN model accuracy
	11.09.2022	Finding the best among the two models and testing it on various faces
	12.09.2022	Correcting errors and improving accuracy
	13.09.2022	Correcting errors and improving accuracy
	14.09.2022	Finalize the model with more accuracy

	Date	Topics learned / Activity carried out / Task completed / Online /E-resourcesaccessed
WEEK 7	15.09.2022	2nd review changes and feedbacks were noted
	16.09.2022	Changes were made and the model was completed
	17.09.2022	Report Writing

Signature of the Student
(with date)

Signature of the faculty guide
(with date