FACIAL EMOTION REOGNITION

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE
Submitted by

PATIL HITESH REDDY
[RA2011003011163]
C GANESH
[RA2011003011170]
SAI HARSHITH YADDALA
[RA2011003011173]
BHEEMA SHYAM KUMAR
[RA2011003011174]

Under the guidance of
Mr. Arulalan V
Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **“FACIAL EMOTION RECOGNITION”** is the bona fide work of **PATIL HITESHREDDY(RA2011003011163),CHAPPI GANESH (RA2011003011170),SAIHARSHITH YADDALA(RA2011003011173),BHEEMA SHYAM KUMAR(RA2011003011174)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE OF EXTERNAL EXAMINER
Mr. V. Arulalan
Assistant Professor
Department of Computing Technologies

SIGNATURE OF INTERNAL EXAMINER
Dr. M. Pushpalatha
HEAD OF THE DEPARTMENT
Professor & Head
Department of Computing Technologies

ACKNOWLEDGEMENT

We express our humble gratitude to Dr. C. Muthamizhchelvan, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology. Dr. T.V. Gopal, for his invaluable support

We wish to thank Dr. Revathi Venkataraman, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We register our immeasurable thanks to our Faculty Advisor, DR. M. ARUNA, Assistant Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, Mr. V. Arulalan Associate Professor, Department of Artificial Intelligence, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his mentorship. He provided us with the freedom and support to explore the research topics of our interest. His passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Artificial Intelligent Department staff and Students, SRM Institute of Science and Technology, for their help during our project. Finally, would like to thank parents, family members, and friends for their unconditional love, constant support and Encouragement.

PATIL HITESH REDDY(RA2011003011163)
GANESH CHAPPI (RA2011003011170)
SAI HARSHITH YADDALA(RA2011003011173)
BHEEMA SHYAM KUMAR(RA2011003011174)

ABSTRACT

Facial emotion recognition is an emerging field of research that involves the development of computer algorithms to automatically detect and interpret human emotions from facial expressions. This technology has many potential applications in fields such as psychology, human-computer interaction, and marketing, among others. The ability to accurately recognize and interpret facial expressions is an essential component of human communication and social interaction. Facial emotion recognition technology seeks to replicate this ability in computers and machines.

The process of facial emotion recognition involves capturing images or video of a person's face, analyzing the facial features and expressions using machine learning algorithms, and identifying the emotional state of the person based on this analysis.

This training process enables the neural network to learn to recognize the subtle differences in facial expressions that are associated with different emotions. There are many challenges associated with facial emotion recognition. One of the biggest challenges is the variability in facial expressions across different cultures and individuals. What may be interpreted as a happy expression in one culture may be interpreted as neutral or even negative in another culture. Another challenge is the variability in facial expressions within an individual.

The same person may express the same emotion differently depending on the situation, their mood, and other factors. Despite these challenges, recent advances in deep learning and computer vision techniques have greatly improved the accuracy and reliability of facial emotion recognition technology

TABLE OF CONTENTS

ABSTRACT iii

TABLE OF CONTENTS iv

LIST OF FIGURES v

ABBREVIATIONS vi

1 INTRODUCTION 7

2 LITERATURE SURVEY 8

3 SYSTEM ARCHITECTURE AND DESIGN 9

4 METHODOLOGY 14

5 CODING AND TESTING 15

6 SREENSHOTS AND RESULTS

6.1 Buzzer and PIR 19

6.2 Ultrasonic and Scarecrow 19

6.3 Soil moisture sensor with water pump 20

6.4 LCD Screen 20

6.5 Whole Circuit 21

6.6 Thingspeak Server 22

7 CONCLUSION AND FUTURE ENHANCEMENT 23

7.1 Conclusion

7.2 Future Enhancement

REFERENCES 24

LIST OF FIGURES

3.1.1	**Architecture block**	9
3.2.1	**Soil moisture sensor**	10
3.2.3	**Temperature Sensor (DHT 11)**	10
3.2.4	**PIR Motion Sensor**	10
3.2.5	**Motor**	11
3.2.6	**Ultrasonic Sensor**	11
3.2.7	**Buzzer**	12
3.2.8	**WIFI MODULE ESP 8266**	13
6.1.1	**Buzzer and PIR simulation on tinkercad**	19
6.2.1	**Ultrasonic sensor and Scarecrow simulation on tinkercad**	19
6.3.1	**Soilmoisture sensor and motor simulation on tinkercad**	20
6.4.1	**LCD Screen with output**	20
6.5.1	**Whole Circuit**	21
6.6.1	**Screenshot of thingspeak server**	22

ABBREVIATIONS

IOT	Internet of Things
PIR	Passive Infrared
LCD	Liquid Crystal Diode
DHT	Distributed hash table
IR	Infra red
UART	Universal Asynchronous Receiver/Transmitter
IDE	Integrated Development Environment

CHAPTER 1

INTRODUCTION

Facial emotion recognition is a rapidly growing field of research and application that involves the use of computer algorithms to analyze and interpret human facial expressions and identify the corresponding emotions.

This technology has numerous potential applications in fields such as video surveillance, healthcare, education, and entertainment, where it can be used to recognize and respond to the emotions of the users or customers.
By analyzing the subtle changes in facial features, such as the position and movement of the eyes, eyebrows, nose, mouth, and chin, facial emotion recognition can provide valuable insights into the emotional states of individuals and facilitate more personalized and effective interactions.

In this context, it is important to develop robust and accurate methods for facial emotion recognition that can handle diverse populations and complex scenarios. This paper outlines a general methodology for facial emotion recognition that can serve as a framework for future research and development in this area.

CHAPTER 2

LITERATURE SURVEY

TITLE	JOURNAL NAME	AUTHOR	DATE OF PUBLICATION	HIGHLIGHTS
FACIAL RECOGNITION USING SVM	INTERNATIONAL JOURNAL OF PATTERN RECOGNITION AND AI	KWANG INKIM JIN HYUN GKIM	NOVEMBER 2021	support vector machines (SVMs), which are known to work well even in high-dimensional space, are used as the face recognizer.Their basic scheme is extended for multiface recognition by adopting one-per-class decomposition
FACIAL EXPRESSION RECOGNITION USING FEATURE EXTRACTION AND CLASSIFICATION TECHNIQUE	INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND APPLICATIONS	J.LETHISIA NITHIYA	FEBRUARY 2020	System-based identification has been a vital area of research in the literature for a long time. The presented work confers a new framework for facial expression recognition from video files by selecting the Gabor features on video frames

Rapid Object Detection using a Boosted Cascade of Simple Features	ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION	Paul Viola Michael Jones	2021	“Integral Image” which allows the features used by our detector to be computed very quickly. “Cascade” which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

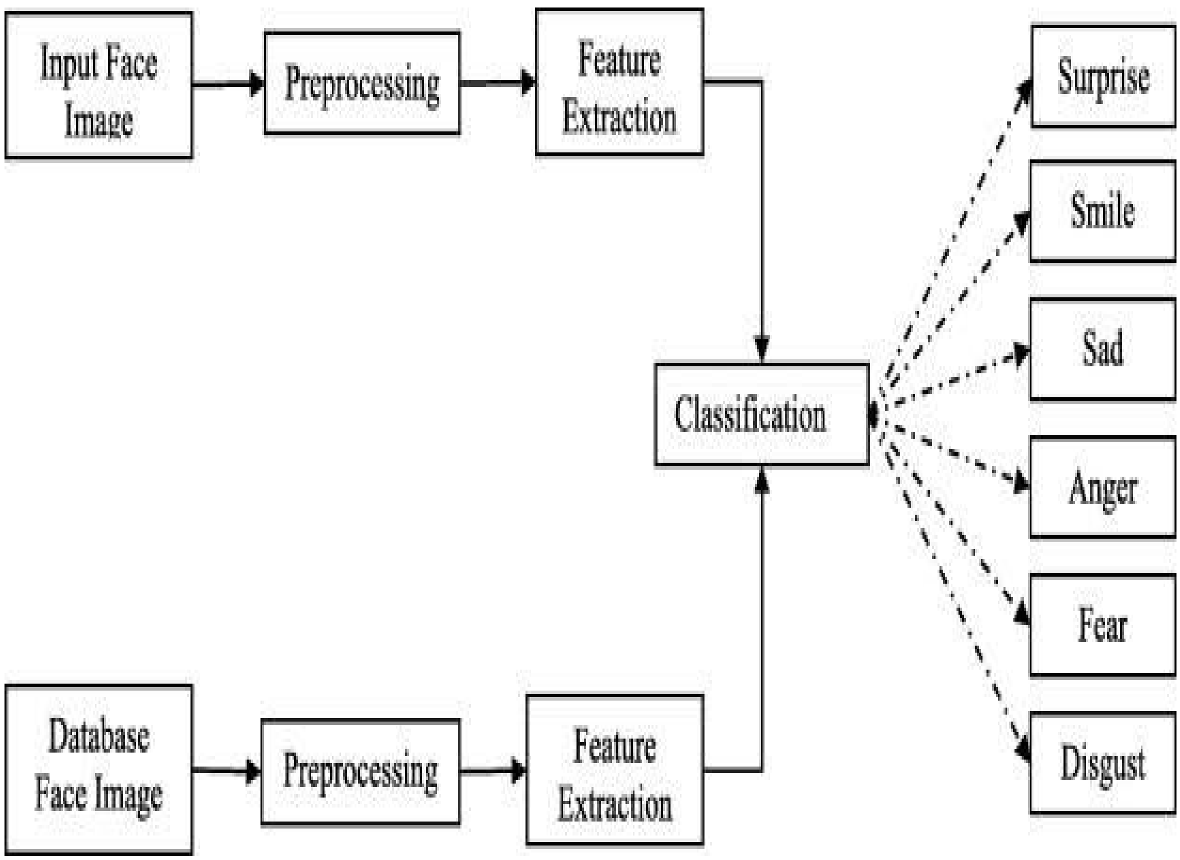


Fig 1.1 :- System Architecture

CHAPTER 4

METHODOLOGY

1. Data collection: Collect a large and diverse dataset of images or videos of human faces expressing different emotions, such as happiness, sadness, anger, fear, surprise, and disgust. The dataset should include a variety of individuals of different ages, genders, and ethnicities, and should be labeled with the corresponding emotion.

2. Preprocessing: Preprocess the data by detecting and aligning the facial landmarks, normalizing the lighting conditions, and cropping the face region to reduce the noise and variability in the images.

3. Feature extraction: Extract features from the preprocessed images that capture the distinctive facial expressions of each emotion. Common features include facial landmarks, such as the positions and angles of the eyes, eyebrows, nose, mouth, and chin, as well as texture features, such as the local binary patterns or Gabor filters.

4. Classification: Train a machine learning or deep learning model on the extracted features to classify the input images into the corresponding emotion categories. Common classification algorithms include Support Vector Machines (SVM), Random Forests, Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN).

5. Evaluation: Evaluate the performance of the trained model using metrics such as accuracy, precision, recall, and F1-score on a validation dataset. Fine-tune the model by adjusting the hyperparameters and optimizing the training process to achieve better results.

CHAPTER 5

CODING AND TESTING

MODULES

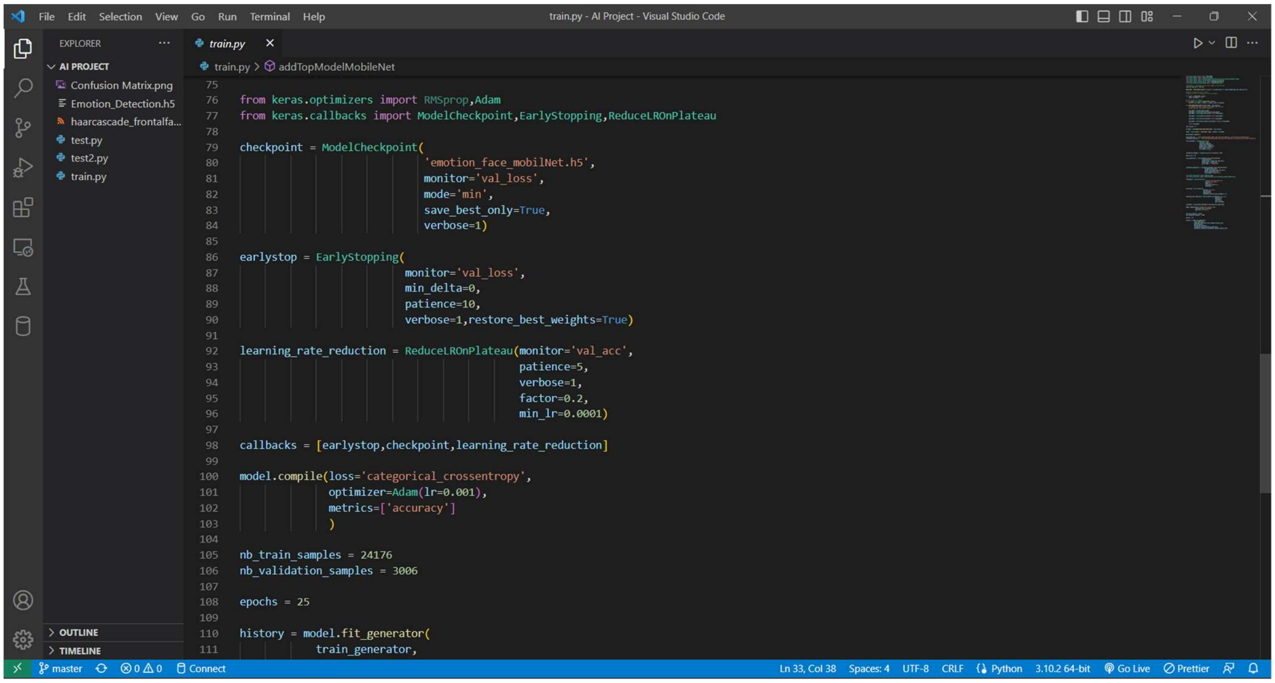
Train.py

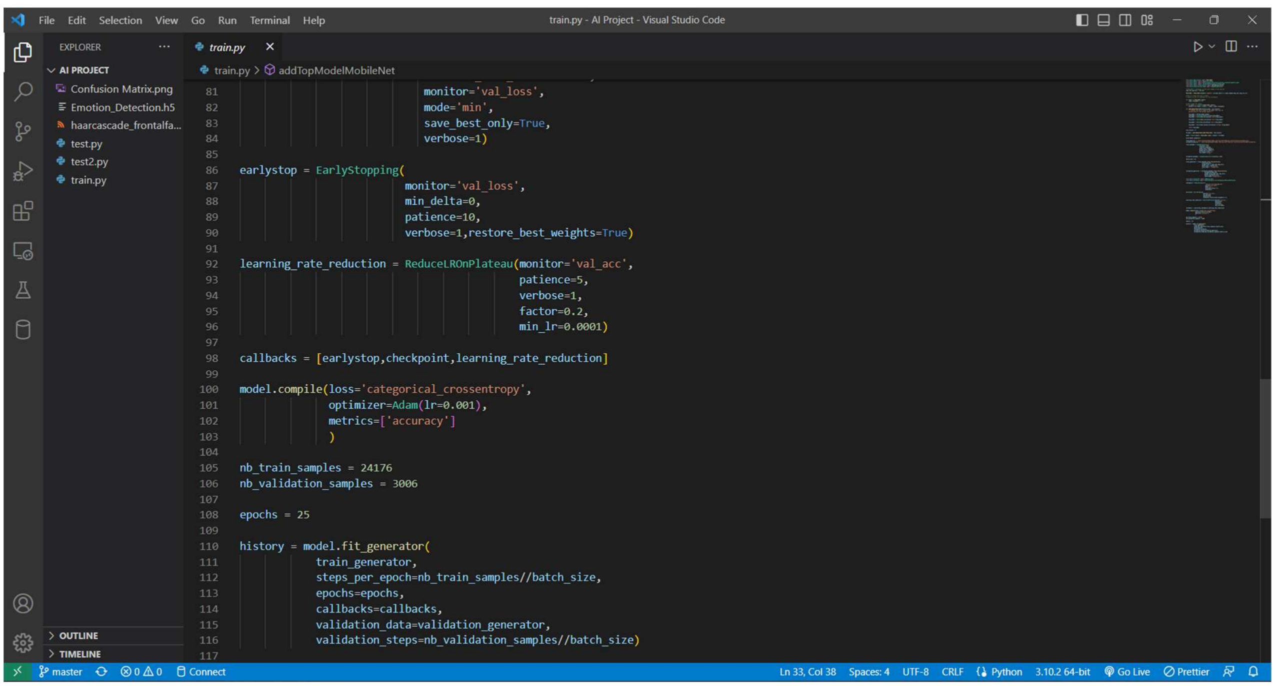
This is a Python code that uses Keras library to build a deep learning model for emotion classification using the MobileNet architecture.
The model is trained on a dataset of facial images that belong to one of five emotion classes (happy, sad, angry, surprised, and neutral).

- The code creates a new model using the input and output of the pre-trained MobileNet model.
- Next, the code defines the directories where the training and validation images are located and creates ImageDataGenerators for both directories. It applies various image augmentations to the training data such as rotation, shifting, and flipping to improve the generalization of the model.
- Then, it sets up various callbacks such as ModelCheckpoint, EarlyStopping, and ReduceLROnPlateau to monitor the training process and save the best model.
- Finally, the code compiles the model with Adam optimizer, categorical_crossentropy loss function, and accuracy metric, and then trains the model using the fit_generator function. The training process is monitored by the defined callbacks.

```python
from keras.applications import MobileNet
from keras.models import Sequential,Model
from keras.layers import Dense,Dropout,Activation,Flatten,GlobalAveragePooling2D
from keras.layers import Conv2D,MaxPooling2D,ZeroPadding2D
from keras.layers.normalization import BatchNormalization
from keras.preprocessing.image import ImageDataGenerator

# MobileNet is designed to work with images of dim 224,224
img_rows,img_cols = 224,224

MobileNet = MobileNet(weights='imagenet',include_top=False,input_shape=(img_rows,img_cols,3))

# Here we freeze the last 4 layers
# Layers are set to trainable as True by default

for layer in MobileNet.layers:
    layer.trainable = True

# Let's print our layers
for (i,layer) in enumerate(MobileNet.layers):
    print(str(i),layer.__class__.__name__,layer.trainable)

def addTopModelMobileNet(bottom_model, num_classes):
    """creates the top or head of the model that will be
    placed ontop of the bottom layers"""

    top_model = bottom_model.output
    top_model = GlobalAveragePooling2D()(top_model)
    top_model = Dense(1024,activation='relu')(top_model)

    top_model = Dense(1024,activation='relu')(top_model)

    top_model = Dense(512,activation='relu')(top_model)

    top_model = Dense(num_classes,activation='softmax')(top_model)

    return top_model
```

```python

num_classes = 5

FC_Head = addTopModelMobileNet(MobileNet, num_classes)

model = Model(inputs = MobileNet.input, outputs = FC_Head)

print(model.summary())

train_data_dir = '/Users/durgeshthakur/Deep Learning Stuff/Emotion Classification/fer2013/train'
validation_data_dir = '/Users/durgeshthakur/Deep Learning Stuff/Emotion Classification/fer2013/validation'

train_datagen = ImageDataGenerator(
                    rescale=1./255,
                    rotation_range=30,
                    width_shift_range=0.3,
                    height_shift_range=0.3,
                    horizontal_flip=True,
                    fill_mode='nearest'
                                   )

validation_datagen = ImageDataGenerator(rescale=1./255)

batch_size = 32

train_generator = train_datagen.flow_from_directory(
                        train_data_dir,
                        target_size = (img_rows,img_cols),
                        batch_size = batch_size,
                        class_mode = 'categorical'
                        )

validation_generator = validation_datagen.flow_from_directory(
                            validation_data_dir,
                            target_size=(img_rows,img_cols),
                            batch_size=batch_size,
                            class_mode='categorical')
```


train.py - AI Project - Visual Studio Code
from keras.optimizers import RMSprop,Adam
from keras.callbacks import ModelCheckpoint,EarlyStopping,ReduceLROnPlateau
checkpoint = ModelCheckpoint(
'emotion_face_mobilNet.h5',
monitor='val_loss',
mode='min',
save_best_only=True,
verbose=1)
earlystop = EarlyStopping(
monitor='val_loss',
min_delta=0,
patience=10,
verbose=1,restore_best_weights=True)
learning_rate_reduction = ReduceLROnPlateau(monitor='val_acc',
patience=5,
verbose=1,
factor=0.2,
min_lr=0.0001)
callbacks = [earlystop,checkpoint,learning_rate_reduction]
model.compile(loss='categorical_crossentropy',
optimizer=Adam(lr=0.001),
metrics=['accuracy']
)
nb_train_samples = 24176
nb_validation_samples = 3006
epochs = 25
history = model.fit_generator(
train_generator,

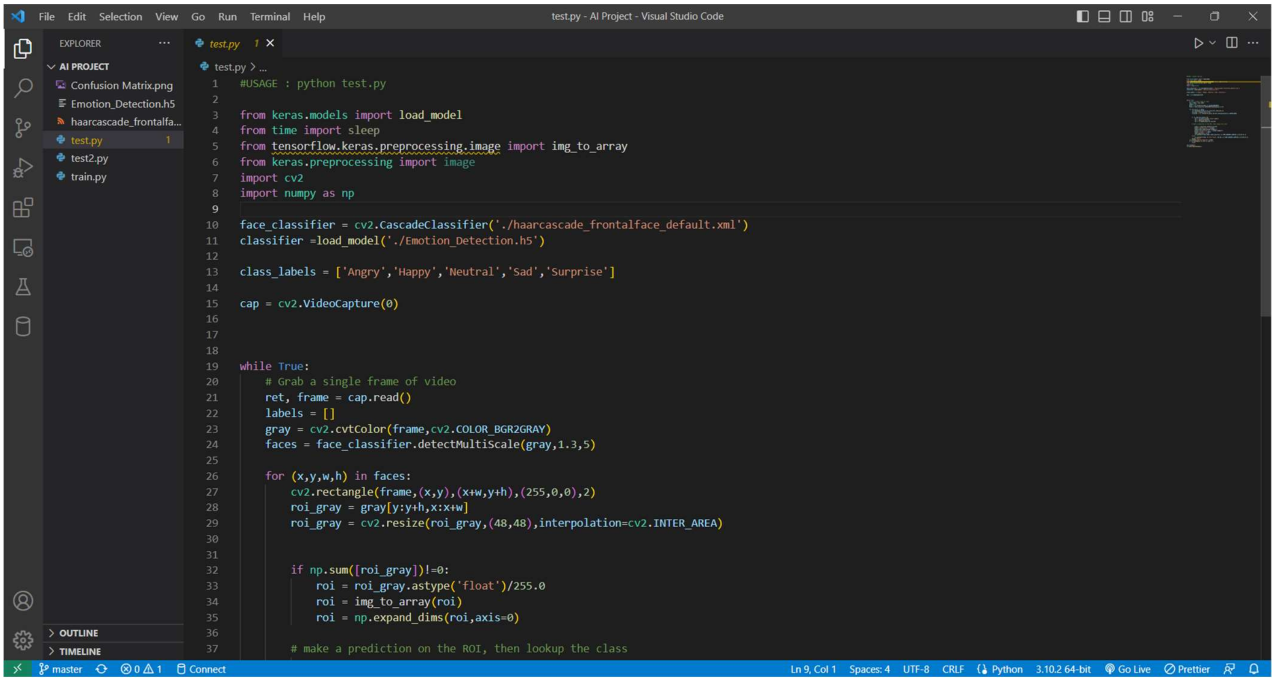

train.py - AI Project - Visual Studio Code
monitor='val_loss',
mode='min',
save_best_only=True,
verbose=1)
earlystop = EarlyStopping(
monitor='val_loss',
min_delta=0,
patience=10,
verbose=1,restore_best_weights=True)
learning_rate_reduction = ReduceLROnPlateau(monitor='val_acc',
patience=5,
verbose=1,
factor=0.2,
min_lr=0.0001)
callbacks = [earlystop,checkpoint,learning_rate_reduction]
model.compile(loss='categorical_crossentropy',
optimizer=Adam(lr=0.001),
metrics=['accuracy']
)
nb_train_samples = 24176
nb_validation_samples = 3006
epochs = 25
history = model.fit_generator(
train_generator,
steps_per_epoch=nb_train_samples//batch_size,
epochs=epochs,
callbacks=callbacks,
validation_data=validation_generator,
validation_steps=nb_validation_samples//batch_size)

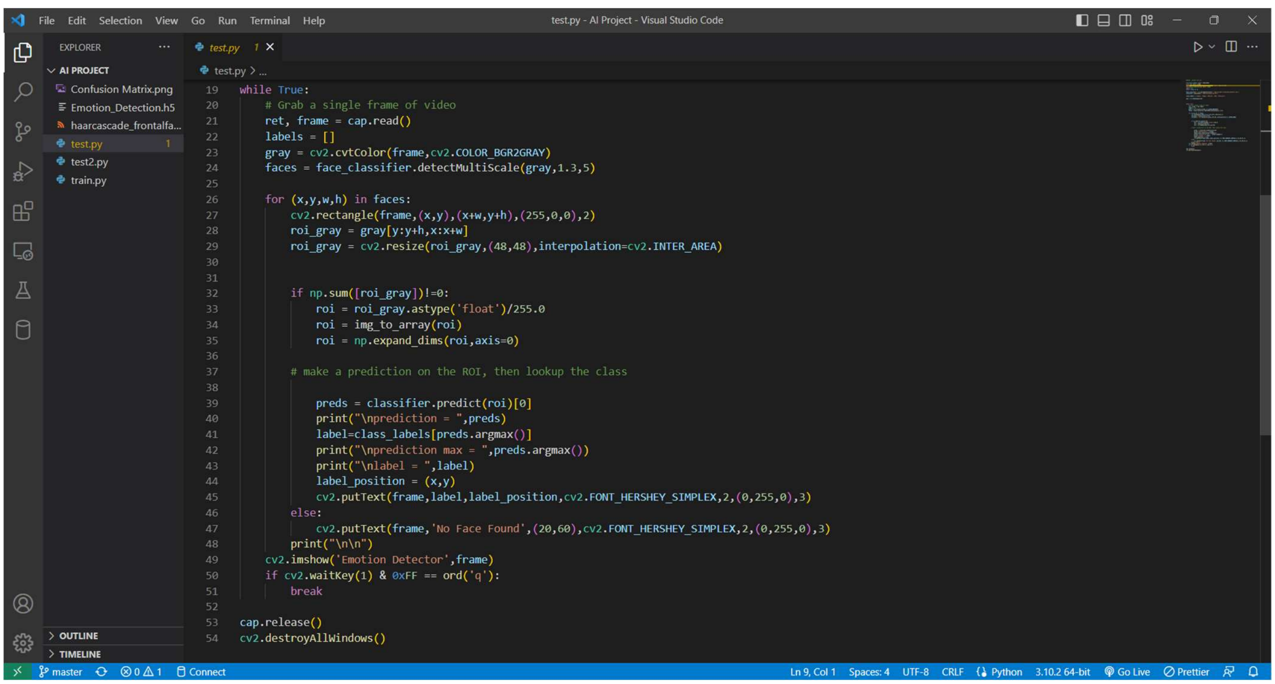
Test.py

Python script for real-time facial emotion detection using our trained Convolutional Neural Network model (Emotion_Detection.h5) and the OpenCV library for face detection.

Overview of the script:

- Import necessary libraries: cv2, numpy, time, keras, and tensorflow.
- Load the pre-trained face classifier model from the haarcascade_frontalface_default.xml file and the emotion detection model from the Emotion_Detection.h5 file.
- Define the emotion class labels for the output predictions.
- Start the video capture from the default camera (index 0) using the cv2.VideoCapture() function.
- Loop through each frame captured from the video stream using the while loop.
- Detect faces in the current frame using the face classifier model and the cv2.CascadeClassifier.detectMultiScale() function.
- For each detected face, resize the region of interest (ROI) to 48x48 pixels and convert it to a numpy array using the cv2.resize() and tensorflow.keras.preprocessing.image.img_to_array() functions, respectively.
- Preprocess the ROI by normalizing the pixel values and adding an extra dimension using the numpy.expand_dims() function.
- Use the pre-trained emotion detection model to make a prediction on the preprocessed ROI using the model.predict() function.
- Get the predicted label from the class labels using the np.argmax() function and display it on the current frame using the cv2.putText() function.
- Display the current frame in a window using the cv2.imshow() function and wait for the 'q' key to be pressed to quit the application.
- Release the video capture and destroy all windows using the cap.release() and cv2.destroyAllWindows() functions, respectively.

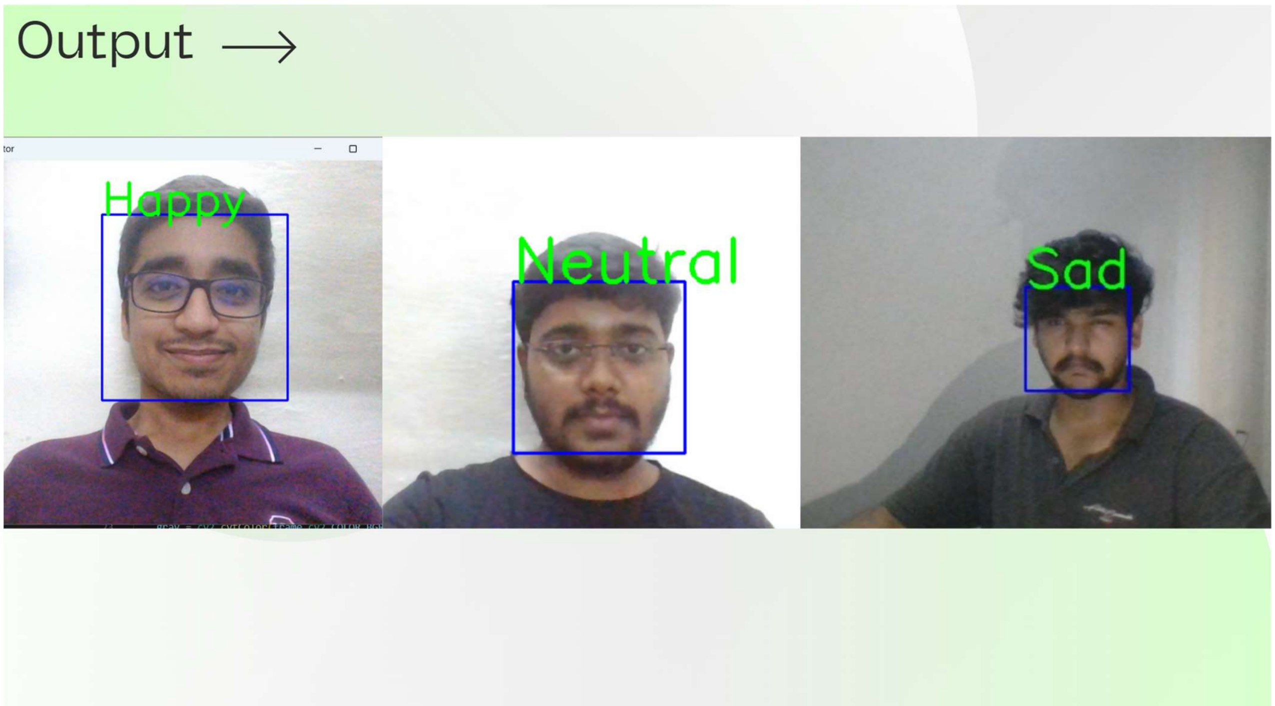

File Edit Selection View Go Run Terminal Help
test.py - AI Project - Visual Studio Code
EXPLORER
AI PROJECT
Confusion Matrix.png
Emotion_Detection.h5
haarcascade_frontalfa...
test.py
test2.py
train.py
OUTLINE
TIMELINE
#USAGE : python test.py
from keras.models import load_model
from time import sleep
from tensorflow.keras.preprocessing.image import img_to_array
from keras.preprocessing import image
import cv2
import numpy as np
face_classifier = cv2.CascadeClassifier('./haarcascade_frontalface_default.xml')
classifier =load_model('./Emotion_Detection.h5')
class_labels = ['Angry','Happy','Neutral','Sad','Surprise']
cap = cv2.VideoCapture(0)
while True:
# Grab a single frame of video
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)
if np.sum([roi_gray])!=0:
roi = roi_gray.astype('float')/255.0
roi = img_to_array(roi)
roi = np.expand_dims(roi,axis=0)
# make a prediction on the ROI, then lookup the class
master
Connect
Ln 9, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.2 64-bit Go Live Prettier


File Edit Selection View Go Run Terminal Help
test.py - AI Project - Visual Studio Code
EXPLORER
AI PROJECT
Confusion Matrix.png
Emotion_Detection.h5
haarcascade_frontalfa...
test.py
test2.py
train.py
OUTLINE
TIMELINE
while True:
# Grab a single frame of video
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)
if np.sum([roi_gray])!=0:
roi = roi_gray.astype('float')/255.0
roi = img_to_array(roi)
roi = np.expand_dims(roi,axis=0)
# make a prediction on the ROI, then lookup the class
preds = classifier.predict(roi)[0]
print("\nprediction = ",preds)
label=class_labels[preds.argmax()]
print("\nprediction max = ",preds.argmax())
print("\nlabel = ",label)
label_position = (x,y)
cv2.putText(frame,label,label_position,cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
else:
cv2.putText(frame,'No Face Found',(20,60),cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
print("\n\n")
cv2.imshow('Emotion Detector',frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
break
cap.release()
cv2.destroyAllWindows()
master
Connect
Ln 9, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.2 64-bit Go Live Prettier

CHAPTER 6

SCREENSHOTS AND RESULTS

OUTPUT



Confusion Matrix

- A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.
- It allows you to visualize the performance of an algorithm by showing the number of correct and incorrect classifications made by the algorithm.
- The confusion matrix is composed of four different values: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

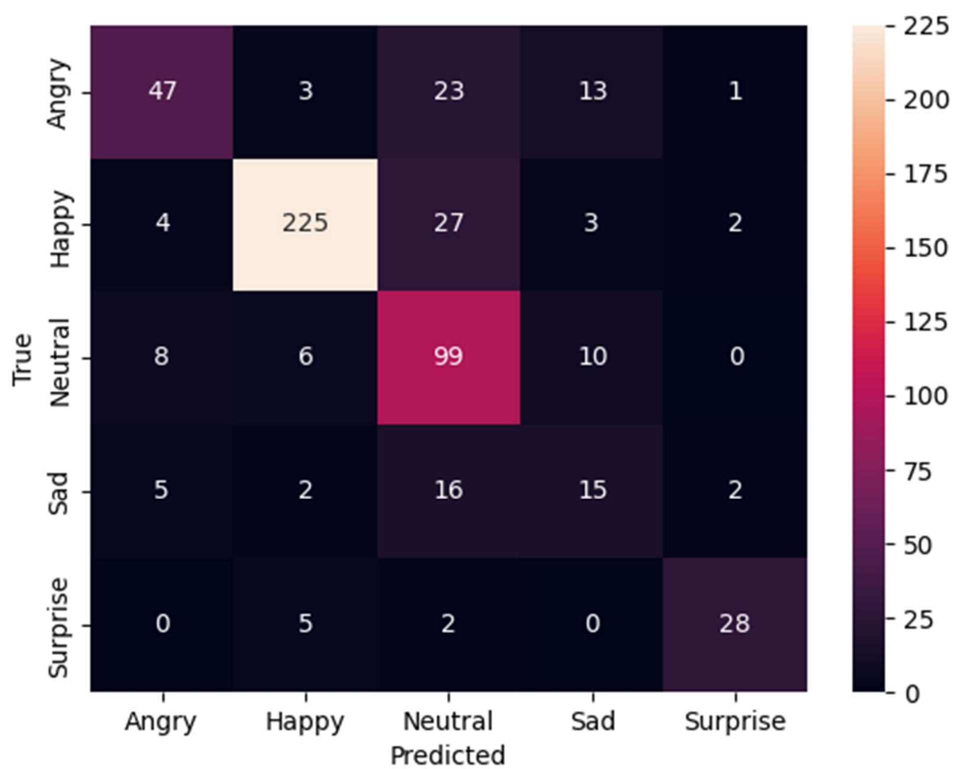


Fig 1.2 matrix table

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, facial emotion recognition technology has the potential to revolutionize many fields, including psychology, human-computer interaction, and marketing. While there are many challenges associated with this technology, recent advances in deep learning and computer vision techniques have greatly improved its accuracy and reliability. The software requirements for facial emotion recognition typically include image or video processing software, computer vision software, facial recognition software, emotion recognition software, and user interface software.
Looking to the future, there are many areas in which facial emotion recognition technology could be enhanced. One area is improving the accuracy and reliability of the technology for recognizing emotions across different cultures and individual variability. This could be achieved through the development of more diverse and representative training datasets, as well as the incorporation of more sophisticated cultural and contextual information into the machine learning algorithms.

Another area for enhancement is the real-time application of facial emotion recognition technology. Real-time application could enable more natural and responsive interactions between humans and machines, and could open up new opportunities for applications in fields such as gaming, virtual reality, and robotics.

In addition, facial emotion recognition technology could be enhanced by incorporating other sources of information, such as voice recognition and physiological signals, into the analysis of emotions. This could provide a more comprehensive and accurate picture of an individual's emotional state, and could enable more personalized and adaptive interactions between humans and machines.

Overall, facial emotion recognition technology has the potential to significantly impact many aspects of human interaction and communication. With continued research and development, this technology could lead to new insights into human emotions and behavior, and could open up new opportunities for innovation and progress in a wide range of fields.

REFERENCES

[1] Li, X., Li, Y., Li, W., & Zhu, C. (2019). Deep convolutional neural networks for facial expression recognition: A survey. Neurocomputing.

[2] Lopes, A. R., Bouwmans, T., & Frélicot, C. (2020). Facial expression recognition using deep convolutional neural networks: A systematic review. Pattern Recognition.

[3] Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2016). AffectNet: A database for facial expression, valence, and arousal computing in the wild. IEEE Transactions on Affective Computin.

[4] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. Proceedings of the British Machine Vision Conferenc.

[5] Sharma, A., & Sharma, N. (2020). Convolutional neural network-based facial expression recognition: A review. International Journal of Computer Application.

[6] Tian, Y., Kanade, T., & Cohn, J. F. (2001). Recognizing action units for facial expression analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[7] Zhang, L., & Wang, S. (2018). Facial expression recognition using convolutional neural networks: A review. Human-centric Computing and Information Sciences.