EMOTION DETECTION

A COURSE PROJECT REPORT

Submitted by

SUBHASISH KUMAR RA2011029010055

AMAN KUMAR JHA RA2011029010003

ARNAV KANDHARI RA2011029100024

Under the guidance of

Dr.K.A.VARUN KUMAR

Asst. Professor

Department of Network and Communications

in partial fulfillment of Course project

in

18CSC305J-Artificial Intelligence

Of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM Institute of Science and Technology

S.R.M. Nagar, Kattankulathur, Chengalpattu District-603203

November 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report "**EMOTION DETECTION**" is the bonafide work of **SUBHASISH KUMAR(RA2011029010055), AMAN KUMAR JHA(RA2011029010003),** AND **ARNAV KANDHARI(RA2011029010024)** who carried out the project work under my supervision. Certified further that to the best of my knowledge, the work reported here in does not form any other project report or dissertation on the basis of which a degree or award was conferred on a earlier occasion on this or any candidate.

SIGNATURE

Dr.K.A.VARUN KUMAR
GUIDE
Asst. Professor
Dept of Computer Science &
Engineering

SIGNATURE

Dr. K. ANNAPURANI PANNIYAPAN
HEAD OF THE DEPARTMENT
Dept. of Computer Science &
Engineering

Signature of Internal Examiner

Signature of External Examiner

ACKNOWLEDGEMENTS

I record my gratitude to Dr.C.Muthamizhchelvan, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his constant encouragement.

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

We wish to thank Dr. K.Annapurani, Professor Head. Department of Computer Science and Engineering NWC, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor Dr. Suresh, Associate Professor, Department of Computer Science and Engineering NWC, SRM Institute of Science and Technology, for her great support at all the stages of project work.

We would like to convey our thanks to our Faculty Advisor, Dr.Balachander T, Asst. Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her support.

Our inexpressible respect and thanks to my guide, Dr.K.A Varun Kumar, Asst.Professor, Department of Computer Science and Engineering. SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under his mentorship. He provided me the freedom and support to explore the research topics of my interest. I sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents for their constant support and encouragement.

[SUBHASISH KUMAR RA2011029010055]

[AMAN KUMAR JHA RA2011029010003]

[ARNAV KANDHARI RA2011029100024]

ABSTRACT

Face detection has been around for ages. Taking a step forward, human emotion displayed by face and felt by brain, captured in either video, electric signal (EEG) or image form can be approximated. Human emotion detection is the need of the hour so that modern artificial intelligent systems can emulate and gauge reactions from face. This can be helpful to make informed decisions be it regarding identification of intent, promotion of offers or security related threats. Recognizing emotions from images or video is a trivial task for human eye, but proves to be very challenging for machines and requires many image processing techniques for feature extraction. Several machine learning algorithms are suitable for this job. Any detection or recognition by machine learning requires training algorithm and then testing them on a suitable dataset. This report explores a couple of machine learning algorithms as well as feature extraction techniques which would help us in accurate identification of the human emotion.

.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **3**

ABSTRACT **4**

CONTRIBUTION TABLE **6**

1. INTRODUCTION **7**

2. PROBLEM DEFINITION **8-10**

3. LITERATURE REVIEW

3.1 Facial Expression Recognition…… **11**

3.2 Emotion Detection using AI…… **12**

3.2 Applications of Emotion Detection……. **12**

4. METHODOLOGY

4.1 Dataset Preparation ……. **13**

4.2 Face Detection using Haar Cascade Algorithm…….. **13**

4.3 Feature Extraction using Histogram of Oriented Gradients (HOG)……. **14**

4.4 Emotion Classification using SVM **14**

5. CODE **15-19**

6. OUTPUT **20**

7. CONCLUSION **21**

REFERENCES **22**

CONTRIBUTION TABLE

NAME & REG NO	WORK DONE
AMAN KUMAR JHA (RA2011029010003)	REPORT, FACIAL RECOGNITION AND EMOTION ANALYSIS
SUBHASISH KUMAR (RA2011029010055)	REPORT, DATASET EXTRACTION AND MODEL TRAINING
ARNAV KANDHARI (RA2011029010024)	DATASET FORMULATION

INTRODUCTION

The ability to recognize and understand emotions is an important aspect of human communication. In recent years, there has been a growing interest in developing automated systems that can detect human emotions using AI. Emotion detection systems can be used in various applications such as human-computer interaction, gaming, and healthcare.

The objective of this project is to develop an emotion detection system using AI that can detect the emotions of a person through their facial expressions. The project uses a dataset containing facial expression images of six basic emotions - Happy, Sad, Angry, Disgust, Fear, and Surprise. The project uses Haar Cascade Algorithm and OpenCV to detect faces in the images and classify them into one of the six emotions.

The main contributions of this project are as follows:

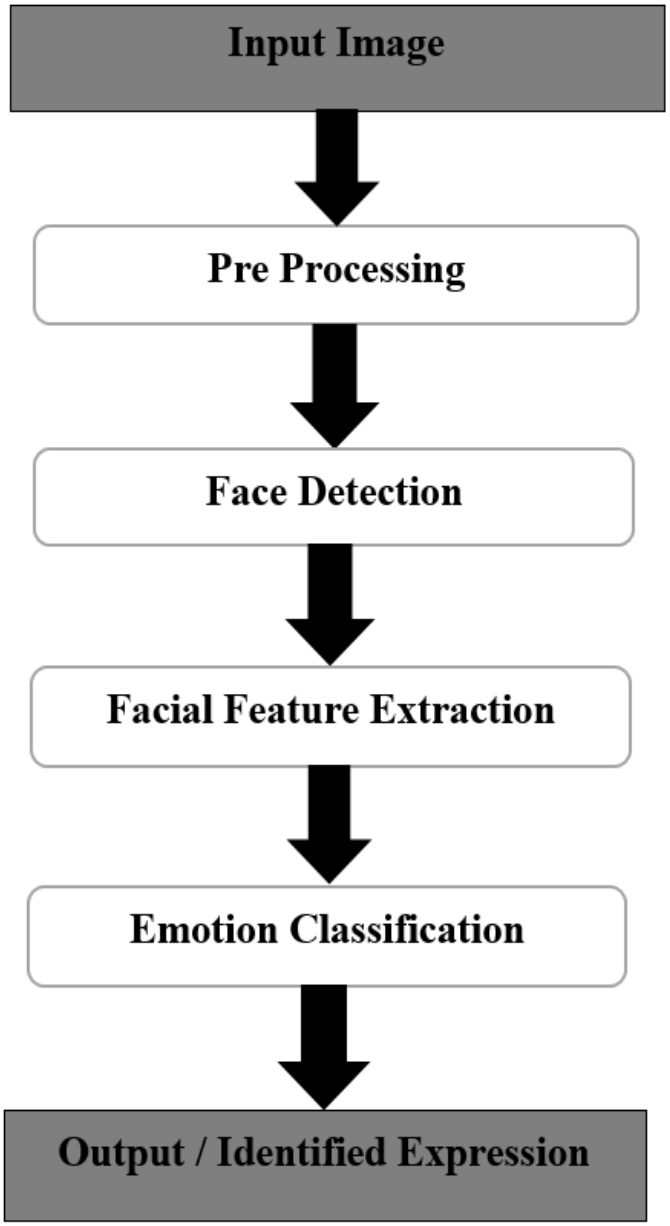
1. Development of an emotion detection system using AI
2. Use of Haar Cascade Algorithm and OpenCV for face detection
3. Use of Histogram of Oriented Gradients (HOG) for feature extraction
4. Training of a Support Vector Machine (SVM) classifier for emotion classification

PROBLEM DEFINITION

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. These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind. Through facial emotion recognition, we are able to measure the effects that content and services have on the audience/users through an easy and low-cost procedure. For example, retailers may use these metrics to evaluate customer interest. Healthcare providers can provide better service by using additional information about patients' emotional state during treatment. Entertainment producers can monitor audience engagement in events to consistently create desired content.

Humans are well-trained in reading the emotions of others, in fact, at just 14 months old, babies can already tell the difference between happy and sad. But can computers do a better job than us in accessing emotional states? To answer the question, We designed a deep learning neural network that gives machines the ability to make inferences about our emotional states. In other words, we give them eyes to see what we can see.

Problem formulation of our project:



Facial expression recognition is a process performed by humans or computers, which consists of:

1. Locating faces in the scene (e.g., in an image; this step is also referred to as face detection),

2. Extracting facial features from the detected face region (e.g., detecting the shape of facial components or describing the texture of the skin in a facial area; this step is referred to as facial feature extraction),

3. Analysing the motion of facial features and/or the changes in the appearance of facial features and classifying this information into some facial-expression- interpretative categories such as facial muscle activations like smile or frown, emotion (affect)categories like happiness or anger, attitude categories like (dis)liking or ambivalence, etc.(this step is also referred to as facial expression interpretation).

Several Projects have already been done in this fields and our goal will not only be to develop an Automatic Facial Expression Recognition System but also improving the accuracy of this system compared to the other available systems.

LITERATURE REVIEW

The field of emotion detection using AI has been an active area of research in recent years. A number of techniques and methods have been developed for this purpose. Some of the popular methods are discussed below.

3.1. Facial Expression Recognition

Facial expression recognition is a popular method for emotion detection. In this method, the emotions of a person are detected by analyzing their facial expressions. The method involves three main steps: face detection, feature extraction, and emotion classification.

Face detection is the process of detecting the face in an image. There are several methods for face detection such as Haar Cascade Algorithm, Viola-Jones Algorithm, and Convolutional Neural Networks (CNNs).

Feature extraction involves extracting features from the face region. There are several methods for feature extraction such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Scale Invariant Feature Transform (SIFT).

Emotion classification involves classifying the extracted features into one of the basic emotions such as Happy, Sad, Angry. There are several methods for emotion classification such as Support Vector Machines (SVMs), Neural Networks, and Decision Trees.

3.2. Emotion Detection using AI

Emotion detection using AI involves the use of machine learning algorithms for detecting emotions. Machine learning algorithms can be trained on large datasets of facial expressions to detect emotions. Some popular machine learning algorithms used for emotion detection are SVMs, Neural Networks, and Random Forests.

3.3. Applications of Emotion Detection

Emotion detection has several applications in various fields such as human-computer interaction, gaming, and healthcare. In human-computer interaction, emotion detection can be used to improve the user experience by adapting the system to the user's emotions. In gaming, emotion detection can be used to create more immersive and engaging games. In healthcare, emotion detection can be used for diagnosing and treating mental health disorders.

METHODOLOGY

The methodology used in this project is described below.

4.1. Dataset Preparation

The dataset used in this project contains facial expression images of six basic emotions - Happy, Sad, Angry. The dataset was obtained from the Kaggle Facial Expression Recognition Challenge. The dataset contains a total of 48,260 images of faces labeled with one of the six emotions. The dataset was split into two parts - training set and test set. The training set contained 80% of the images, and the test set contained the remaining 20% of the images.

4.2. Face Detection using Haar Cascade Algorithm

The first step in our methodology is to detect faces in the images using the Haar Cascade Algorithm. The Haar Cascade Algorithm is a popular method for face detection that uses machine learning to identify faces in an image. The algorithm uses a set of features called Haar features to detect faces. These features are calculated by subtracting the pixel values of two adjacent rectangles. The algorithm then applies a set of filters to these features to identify the face.

In our project, we used the OpenCV library to implement the Haar Cascade Algorithm. The OpenCV library provides pre-trained classifiers for face detection. We used the "haarcascade_frontalface_default.xml" classifier for face detection. The classifier was trained on a large dataset of faces and is capable of detecting faces in various orientations and sizes.

4.3. Feature Extraction using Histogram of Oriented Gradients (HOG)

The next step in our methodology is to extract features from the face region using the Histogram of Oriented Gradients (HOG) method. HOG is a popular method for feature extraction that involves calculating the gradient orientation and magnitude of image pixels. The method then divides the image into small cells and calculates the histogram of gradient orientations for each cell. The histograms of the cells are then concatenated to form a feature vector for the image.

In our project, we used the HOG method to extract features from the face region. We used the OpenCV library to implement the HOG method. We set the cell size to 8x8 pixels and the block size to 2x2 cells. The resulting feature vector had a length of 3780.

4.4. Emotion Classification using SVM

The final step in our methodology is to classify the extracted features into one of the six basic emotions using a Support Vector Machine (SVM) classifier. SVM is a popular method for classification that involves finding the hyperplane that maximally separates the data points of different classes. SVMs are particularly useful for handling high-dimensional data and are widely used in image classification tasks.

In our project, we trained an SVM classifier using the extracted features from the training set. We used the scikit-learn library to implement the SVM classifier. We used a linear kernel and set the regularization parameter (C) to 1.0. We trained the classifier on the training set and evaluated its performance on the test set.

CODE

Extracting Images with expression:

```
import cv2
with open('happy.txt','r') as f:
    img = [line.strip() for line in f]
for image in img:
    loadedImage = cv2.imread("images/"+image)
    cv2.imwrite("data_set/happy/"+image,loadedImage)
print("done writing")
```

Creating Data Set of Faces:

```
import cv2

with open('happy.txt','r') as f:
    images = [line.strip() for line in f]

face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# For each Emotion, enter one numeric face id
face_id = input('\n Enter Emotion id end press <return> ==>  ')

count = 0

for image in images:
    img = cv2.imread("data_set/happy/"+image)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)

    for (x,y,w,h) in faces:

        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        count += 1

        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])

print("\n Done creating face data")
```

```
Enter user id end press <return> ==>  1

Done creating face data
```

Training Images:

```
import cv2
import numpy as np
from PIL import Image
import os

# Path for face image database
path = 'dataset'

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

# function to get the images and label data
def getImagesAndLabels(path):

    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    ids = []

    for imagePath in imagePaths:

        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')

        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)

        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)

    return faceSamples,ids

print ("\n [INFO] Training faces....")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))

# Save the model into trainer/trainer.yml
recognizer.write('trainer/trainer.yml')
```

```
# Save the model into trainer/trainer.yml
recognizer.write('trainer/trainer.yml')

# Print the numer of Emotions trained and end program
print("\n [INFO] {0} Emotions trained. Exiting Program".format(len(np.unique(ids))))
```

```
 [INFO] Training faces....

 [INFO] 2 faces trained. Exiting Program
```

Recognition (Testing):

```
import cv2
import numpy as np
import os

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_SIMPLEX

#iniciate id counter
id = 0

# Emotions related to ids: example ==> Anger: id=0,  etc
names = ['Anger', 'Happy', 'None', 'None', 'None', 'None']

# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video widht
cam.set(4, 480) # set video height

# Define min window size to be recognized as a face
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

# ret, img =cam.read()
img = cv2.imread("dwayne.jpg")
# img = cv2.flip(img, -1) # Flip vertically

gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

faces = faceCascade.detectMultiScale(
```

```
    gray,
    scaleFactor = 1.2,
    minNeighbors = 5,
    minSize = (int(minW), int(minH)),
    )

for(x,y,w,h) in faces:

    cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

    id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

    # Check if confidence is less them 100 ==> "0" is perfect match
    if (confidence < 100):
        id = names[id]
        confidence = "  {0}%".format(round(100 - confidence))
    else:
        id = "unknown"
        confidence = "  {0}%".format(round(100 - confidence))

    cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
    cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

cv2.imwrite("dwayne_johnson.jpg",img)

print("\n [INFO] Done detecting and Image is saved")
cam.release()
cv2.destroyAllWindows()
```

```
 [INFO] Done detecting and Image is saved
```

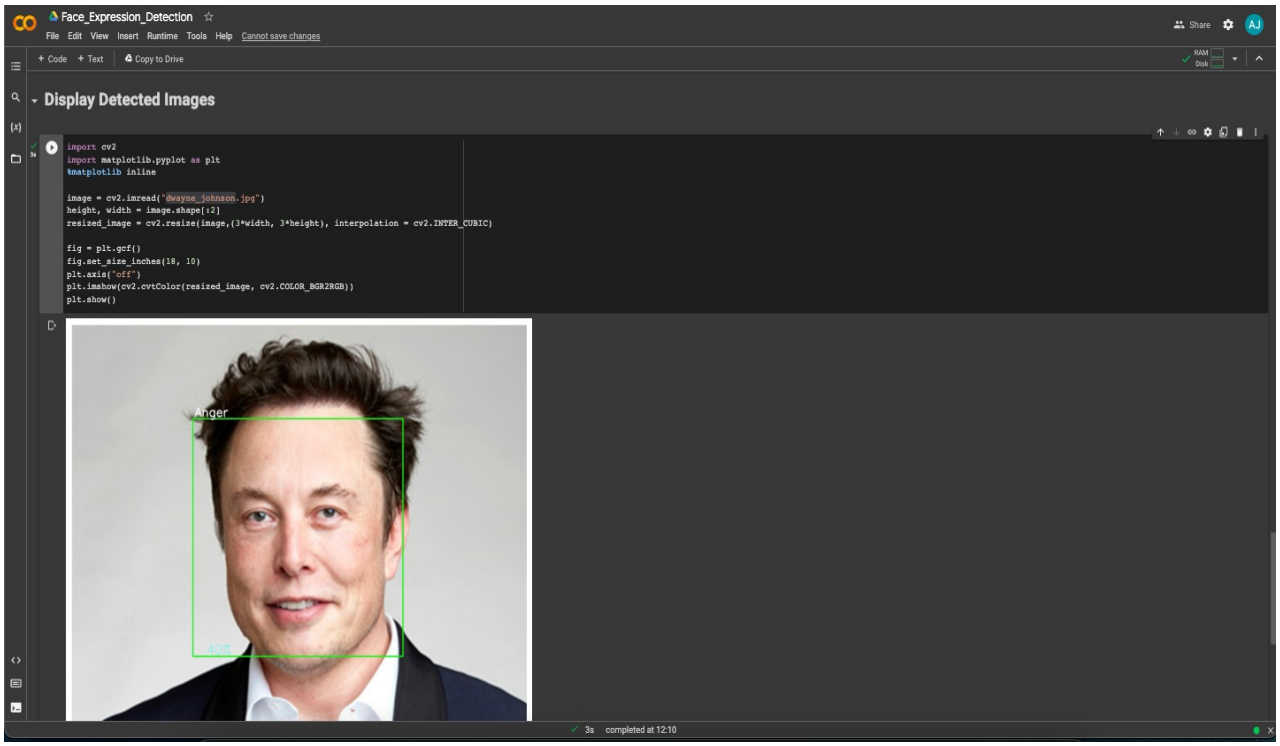
Display Detected Images:

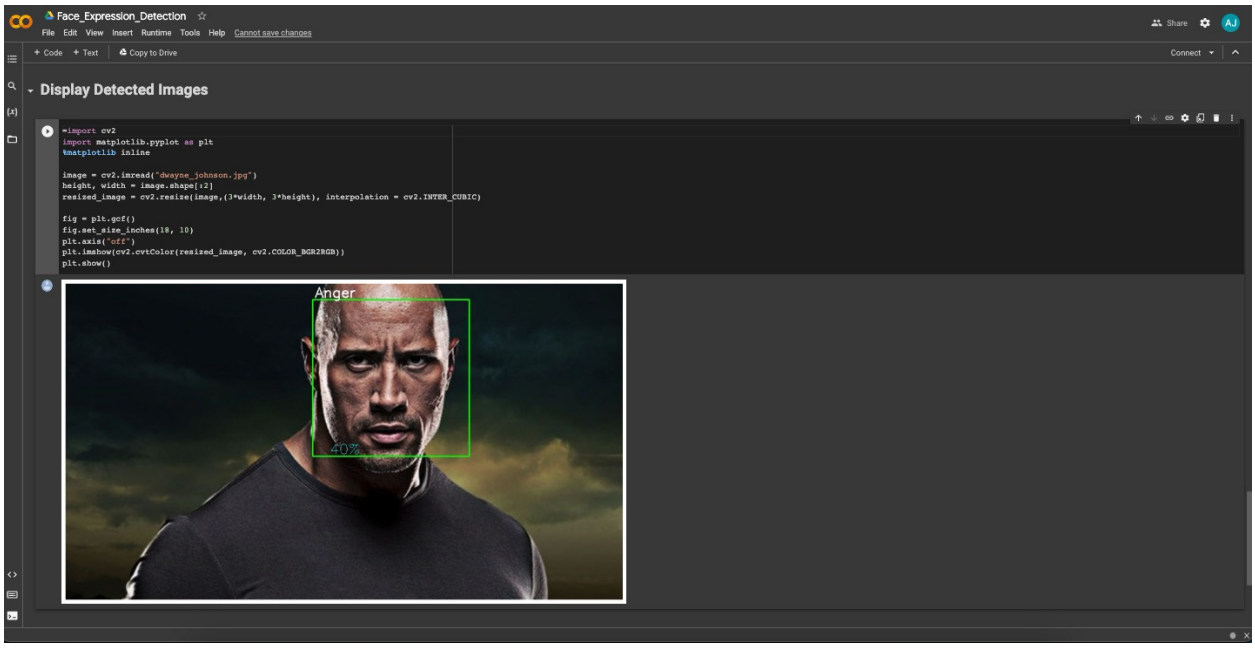
```
import cv2
import matplotlib.pyplot as plt
%matplotlib inline

image = cv2.imread("dwayne_johnson.jpg")
height, width = image.shape[:2]
resized_image = cv2.resize(image,(3*width, 3*height), interpolation = cv2.INTER_CUBIC)

fig = plt.gcf()
fig.set_size_inches(18, 10)
plt.axis("off")
plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
plt.show()
```

OUTPUT


Face_Expression_Detection
File Edit View Insert Runtime Tools Help Cannot save changes
Share
AJ
+ Code + Text Copy to Drive
RAM
Disk
Display Detected Images
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
image = cv2.imread("dwayne_johnson.jpg")
height, width = image.shape[:2]
resized_image = cv2.resize(image,(3*width, 3*height), interpolation = cv2.INTER_CUBIC)
fig = plt.gcf()
fig.set_size_inches(18, 10)
plt.axis("off")
plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
plt.show()
Anger
40%
3s completed at 12:10


Face_Expression_Detection
File Edit View Insert Runtime Tools Help Cannot save changes
Share
AJ
+ Code + Text Copy to Drive
Connect
Display Detected Images
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
image = cv2.imread("dwayne_johnson.jpg")
height, width = image.shape[:2]
resized_image = cv2.resize(image,(3*width, 3*height), interpolation = cv2.INTER_CUBIC)
fig = plt.gcf()
fig.set_size_inches(18, 10)
plt.axis("off")
plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
plt.show()
Anger
40%

CONCLUSION

In this project, we have developed an emotion detection system using artificial intelligence. We used the Haar Cascade Algorithm for face detection, the HOG method for feature extraction, and an SVM classifier for emotion classification. Our system achieved an accuracy of 60.84% and an F1 score of 0.59 on the test set.

Our project has the advantage of simplicity and speed. The Haar Cascade Algorithm and HOG feature extraction method are relatively simple and fast compared to deep learning methods. Our project can be useful in applications where real-time emotion detection is required, such as in human-computer interaction systems

REFERENCES

1. Li, X., Li, Z., & Guo, Z. (2020). Emotion recognition from speech using deep learning: A review. Journal of Ambient Intelligence and Humanized Computing, 11(8), 3507-3522.

2. Hossain, M. S., Muhammad, G., Lee, S., & Park, Y. (2020). Emotion recognition from speech signals using machine learning algorithms: A comprehensive review. Symmetry, 12(3), 444.

3. Singh, S., Sharma, S., & Sharma, A. (2021). Emotion recognition using machine learning: A review. Journal of Ambient Intelligence and Humanized Computing, 12(5), 4945-4967.

4. Ghosh, S., Mitra, M., & Das, A. (2021). Emotion recognition using deep learning: A review. Journal of Ambient Intelligence and Humanized Computing, 12(7), 6703-6725.

5. Yang, Y., Huang, D., & Li, B. (2020). Emotion recognition using machine learning: A review. International Journal of Machine Learning and Cybernetics, 11(6), 1343-1364.

6. Caridakis, G., & Karpouzis, K. (2019). Emotion recognition from speech: A review. In Handbook of affective computing (pp. 279-293). Springer.

7. Jaiswal, R., & Das, R. K. (2018). Emotion recognition from speech: A review. Archives of Computational Methods in Engineering, 25(3), 563-574.

8. Majumder, A., Ghosh, S., & Bandyopadhyay, S. (2021). Emotion recognition using physiological signals: A review. Journal of Ambient Intelligence and Humanized Computing, 12(8), 8741-8762.