RECOGNITION OF FACE EMOTION IN REAL TIME USING MATLAB

Hunny Pahuja[1], Samreen Siddiqui[2], Tanya Varshney[3], Satwik Pal[4] and Amit Kumar Yadav[5]

[1,2,3,4,5]Department of Electronics and Communication Engineering,

KIET Group of Institutions, Delhi-NCR, Ghaziabad, Uttar Pradesh, India

ABSTRACT

Automatic face expression recognition has recently attracted a lot of interest because of the numerous applications it offers in a wide range of industries. This study aims to identify the emotion that a person's face communicates from a grayscale image of their face. The percentage of photos correctly labelled for each emotion will serve as our evaluation criterion and show which emotions are more accurately recognized than others. We can help those who require medication by using this technology. In the subject of automation, this technology can be utilized to understand human emotions. With the use of this technology, vehicles may alert the driver if they are feeling tired, track students who are participating in virtual learning, and more. The person's image is taken, and it is then compared to the images that are already in our database, to complete the task. This system is based on Support Vector Machines (SVM) and Linear Binary Patterns (LBP). Using LBP, its features are extracted and then classified using the SVM algorithm. It is then trained using a learning model and stored on the device. Now for the testing set, whenever an image is taken from a camera, it uses the same steps for extracting its features and classifying it. The results are now obtained by comparing this image to the images kept in our training set.

Keywords:

Facial Expression Recognition (FER), Local Binary Pattern (LBP), Support Vector Machine (SVM)

1. INTRODUCTION

The process of identifying human emotions (such as happiness) on the face is known as facial emotion identification. Happiness is the most significant emotion that each person expresses. MATLAB is the best tool for identifying facial expressions of emotion. Face emotion recognition, or FER, is a well-known abbreviation. The quality of human-machine connections will increase if computers and other everyday technology devices can accurately read facial expressions. Facial expressions can be changed, thus in order to achieve high accuracy, we will be looking for micro expressions. Micro expressions are ones that only last for a very little period of time. They are undetectable to the observers in the area. This happens every single half-second. The same emotion is consistently shown in micro expressions. However, they are related because they make use of emotions to make someone appear to be lying or concealing something, which reveals their true nature (a feeling). The micro expressions are challenging to identify and comprehend. Learning about micro expressions will help you understand the emotion of recognizing others more deeply. Human faces have traits that enable us to distinguish between various expressions.

Human faces have a range of logical features, such as eyes, brows, mouths, noses, and so forth, as you are undoubtedly already aware. These traits are used by the facial expression recognition system to precisely identify the emotion that a person has shown. A statement's emotional impact might be either positive or negative. Finding the right emotion through facial expressions is crucial. Utilizing methods from computer vision and machine learning, the facial recognition system models and categorizes facial traits acquired from images and videos. Face identification algorithms map and extract facial characteristics, then compare them to a database of recognized faces to determine the best match. This technology is applicable in a wide range of circumstances. By identifying and minimizing fake insurance claims, identifying thieves, spotting politically astute arrogances, spotting low-energy drivers, and installing fraud prevention systems, we can assist this technology. We can also use it for public safety purposes, such as checking crime scene footage for criminal intent or looking for terrorist activity in populated regions. It can also be used to track student engagement in online courses, keep tabs on patients' health, and assess consumers' moods. The latest version of MATLAB, R2022b win64, is what we are utilizing for this. The following list includes the main MATLAB toolboxes for recognizing the emotions seen on people's faces: Toolboxes for "Deep Learning," "Machine Learning," "Image Processing," and "Image Acquisition."

2. LITERATURE REVIEW

The difficult subject of automatic person recognition [1] has attracted a lot of interest lately since it has so many uses in so many different industries. Facial recognition is one of those challenging challenges, and as of now, there is no technology that can provide a trustworthy response in every situation. A novel method for recognizing human faces is presented in this research. This research presents a novel face recognition method that combines a SOM-based classifier with features extracted from DCT coefficients. A database of 25 face photographs with five subjects and five images of each subject's face in a variety of emotions was used to test the system in MATLAB. During around 850 epochs of training, the system achieved an 81.36% recognition rate for 10 consecutive trials. When compared to conventional DCT feature extraction methods, the method's computational requirements are drastically lowered by the reduced feature space mentioned for experiment 2 above. This method's primary benefits are its quick processing ability and low computing demands, both in terms of memory utilisation and speed.

Our technology is therefore highly adapted for real-time, low-cost hardware implementation. There aren't any commercial applications of this method at the moment. It is feasible, nevertheless, that a useful SOM-based face recognition system will someday be developed.

We suggested a precise and quick emotion recognition algorithm [2] in this paper. To quickly identify skin color, color and feature-based detections were used together with carefully chosen candidate blocks. This paper's main contribution is that the suggested method can identify image edges, and from those edges, distances between different features are determined using Euclidean distance formulae. For each image portraying a distinct emotion, this distance varies. The classification of emotions is based on this gap. The suggested strategy can be used for hardware implementation in further development. The suggested method's straightforward structure makes it appropriate for hardware implementation in order to produce extremely high performance and low power systems.

One of the most popular uses of machine vision in recent years is the identification of facial emotions. It can be used for machine-human interaction, entertainment, and security (HMI). Emotion recognition frequently makes use of scientifically-based image processing, audio processing, gesture signal processing, and physiological signal processing.

The automatic methods for recognizing face expressions [3] and different research challenges are described._In essence, these systems combine feature extraction, face recognition, and categorization. These processes are included in the architecture of a face detection and input for a facial expression recognition system, normalization, face extraction, classification, and output._A higher recognition rate can be achieved using a variety of ways. Techniques with a higher rate of recognition perform better. Owing of the muddled physical and psychological aspects of emotions that are connected to the unique traits of each person, emotion identification through facial expression is a problem that affects everyone and poses challenges.

Facial expression recognition can be used to identify human emotions. With the aid of the captured series of photos, the proposed system recognized the object. In order to efficiently extract feature points, it recognizes faces in the collected photos. The algorithm successfully assigns each of the photos to one of the six universal emotions. The system's classifier correctly produces the desired results.

In this article [4], we present a system that uses facial expressions to automatically identify human emotions. This research and the procedures employed are more appropriate because they currently employ the most successful facial recognition methods, which decrease response latency. Live streaming, Skin Color Segmentation, Face Detection, Eye Detection, Lip Detection, Longest Binary Pattern, Bezier Curve Algorithm, Emotion detection, Database, and Output display are the ten fundamental components of emotion recognition that are presented. The Cubic Bezier Curve Implementation, which is more versatile and emerges as the most important in a number of diverse industries, including robotics, computer graphics, automation, and animation, also advances the emotion detection approach. The method recognizes the basic six emotional moods and is useful for faces of different shapes, complexions, and skin tones.

The proposed FEER-HRI system [5] is structured as a four-layer system framework and is based on facial expression emotion recognition. The FEER-HRI technology gives the robots the ability to create facial expressions in order to adapt to human emotions in addition to recognizing human emotions. Robotic facial expressions are exhibited on an LED screen that is built into the robots and are represented by straightforward cartoon symbols that are understandable to humans. The four scenarios in the experiment on human-robot interaction include scene simulation, home service, directing, and entertainment. In these situations, fluid communication is achieved via human facial expression recognition and robot face expression production within two seconds. The FEER-HRI system has potential applications in a variety of areas, including safe driving, smart homes, and home services.

3. PROPOSED SYSTEM

3.1 Software and Hardware Requirement

3.1.1 Software Requirement

The latest version of MATLAB, R2022b win64, is what we are utilizing for this. The following list includes the main MATLAB toolboxes for recognizing the emotions seen on people's faces:

- Toolboxes for "Deep Learning"
- Toolboxes for "Machine Learning"
- Toolboxes for "Image Processing"
- Toolboxes for "Image Acquisition"

3.1.2 Hardware Requirement

The hardware requirement for this project is a fluent working laptop with a minimum of 4Gb RAM and a web camera.

3.2 Features for Face Emotion Recognition

The features that are placed on human faces serve as indicators of expressions. As you are aware, human faces have a variety of logical features, including eyes, brows, mouths, noses, and so forth. These characteristics aid the facial emotion recognition system in accurately identifying the emotion displayed by a person.

The association between facial expressions and emotions is seen in the subsequent Table 3.2.1:

Table 3.2.1:

Emotion	Increase the Probability	Decrease the Probability
Happy	Smile	Brows Furrow Brows Raise
Anger	Brow Furrow Lids Tight Eyes Wide Face Raise	Brow Raise Smile Rise Inner Brows
Surprise	Inner Brows Brows Raise Eyes Furrow Jaws Drop	Brows Furrow
Disgust	Nose Wrinkling Upper Lip Curl	Smile Lips Stuck
Sadness	Inside Brows Raise Brows Furrow Lip Corner Depressed	Brows Raise Eyes Wide Mouth Open Lips Pressed Lips Stuck Smile
Fear	Inner Brows Raise Brows Furrow Eyes Wide Lips Stretch	Brows Raise Lip Corner Depressed Jaws Drop Smile

4. METHODOLOGY:
STEPS FOR FACE EMOTION RECOGNITION

- Image selection
- Adding selected images into the database
- Face expression recognition
- Feature extraction using LBP
- Classification using SVM
- Database information
- Removal of database
- Program information
- Source code of the process
- Exit

4.1 Pre-Processing Steps

i. **Code to read an image**
rgbImage = imread(“filename.fmt");

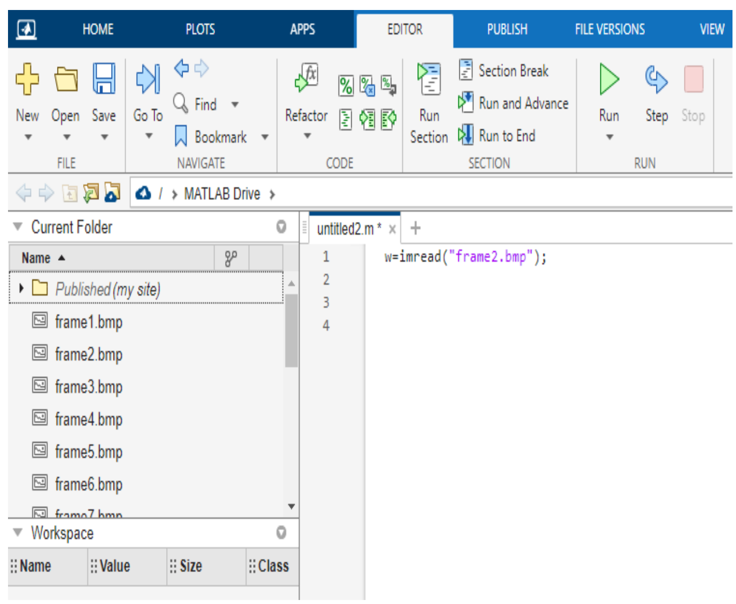


Fig. 4.1.1 Reading an Image

ii. **Code to write an image**
imwrite(img);

iii. **Code for gray scale conversion**
grayimage = rgb2gray(rgbimage);
imshow(grayimage)

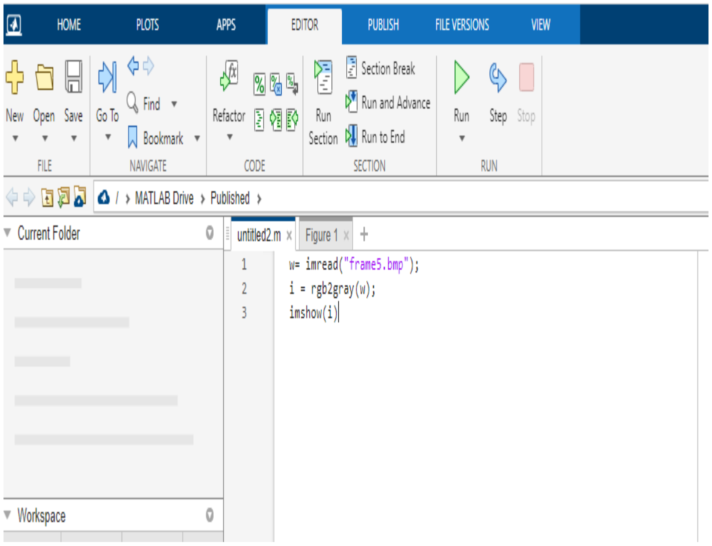


Fig. 4.1.2 Gray Scale Conversion Code

Gray Image Result:

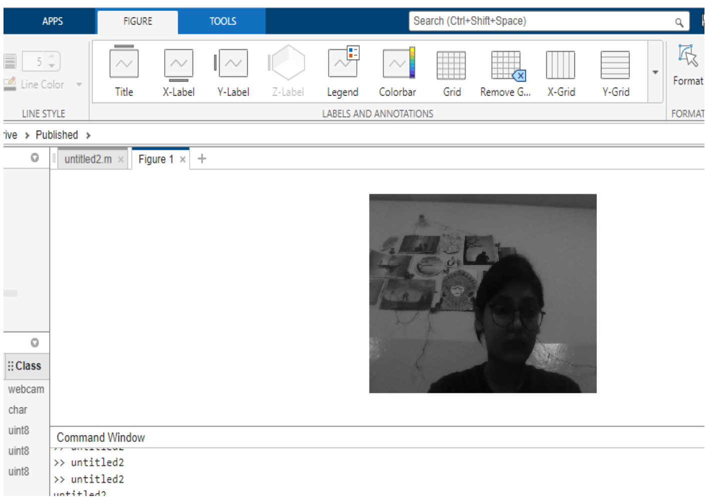


Fig. 4.1.3 Gray Scale Conversion Result

iv. Code to display an image

w=imread(img);

imshow(w)

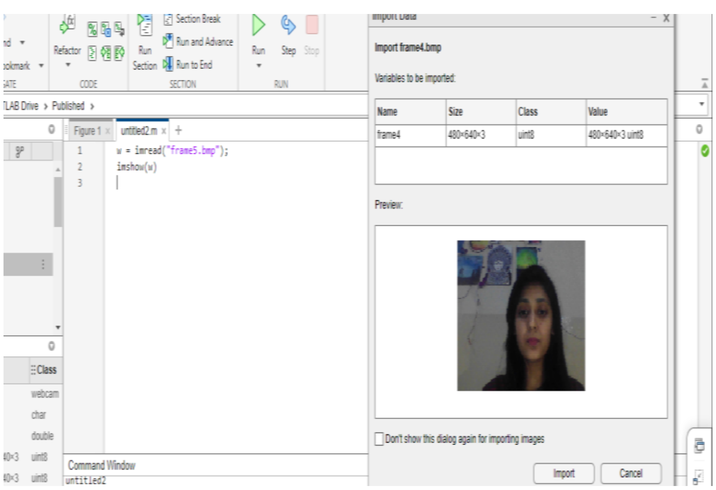


Fig 4.1.4 Image Display

v. Code for reading pixels

i=imread(image);

img=imshow(image)

<u>impixelinfo(img);</u>

Image 1:

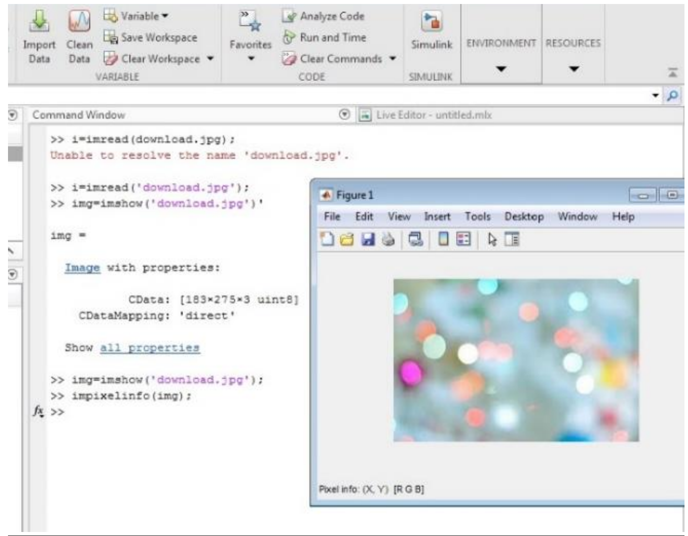


Fig. 4.1.5 impixelinfo() Command

(With Cursor Movements…)

Image 2:

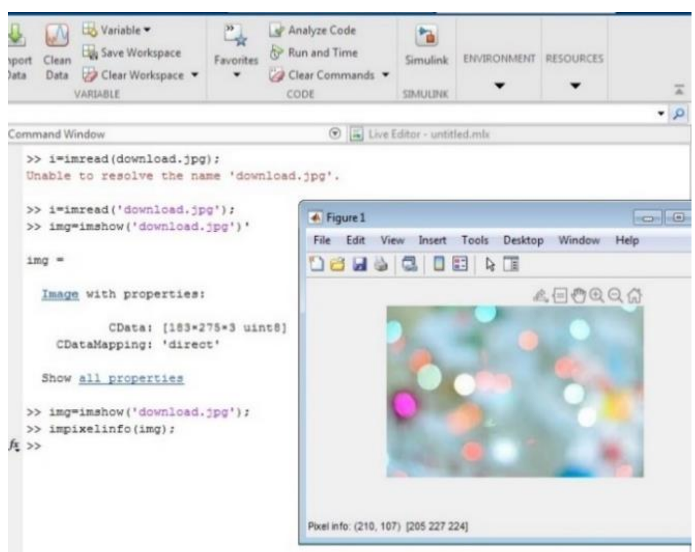


Fig. 4.1.6 impixelinfo() Command with Cursor Movement

Pixels check when person sitting on chair:

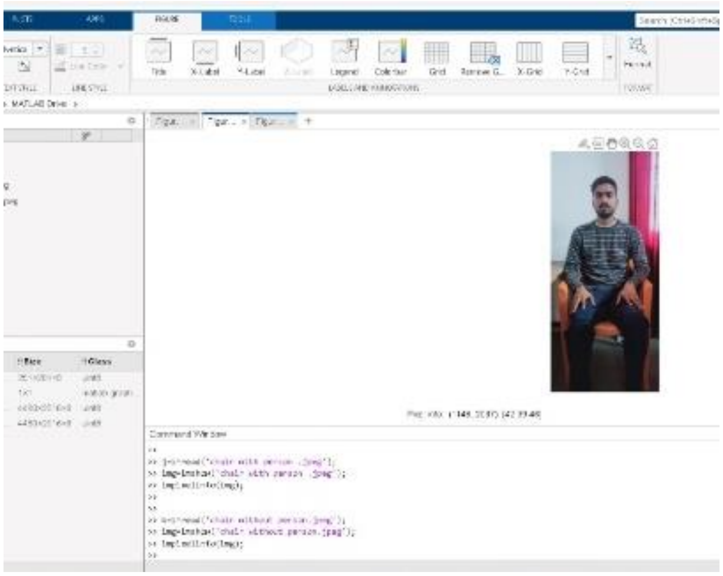


Fig. 4.1.7 Pixel with a Person Seated on Chair

Pixels check when person not sitting on chair:

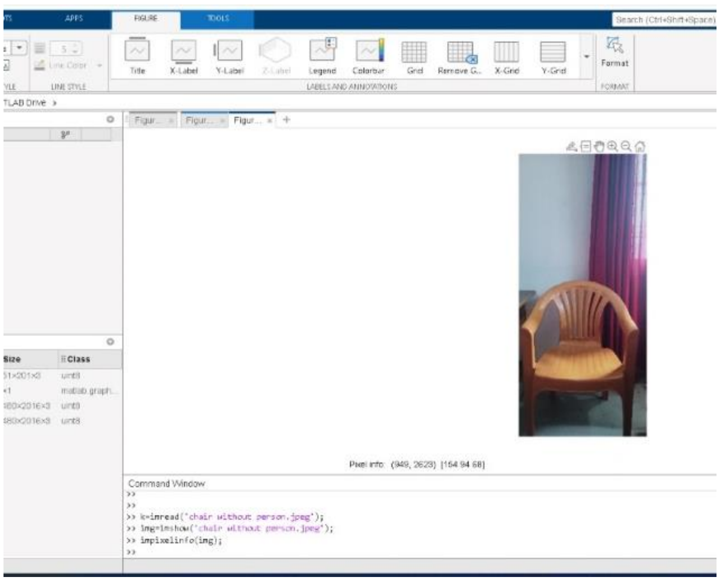


Fig. 4.1.8 Pixel when an Empty Chair

4.2 Experiments

i. How to snap images from webcam

Code:

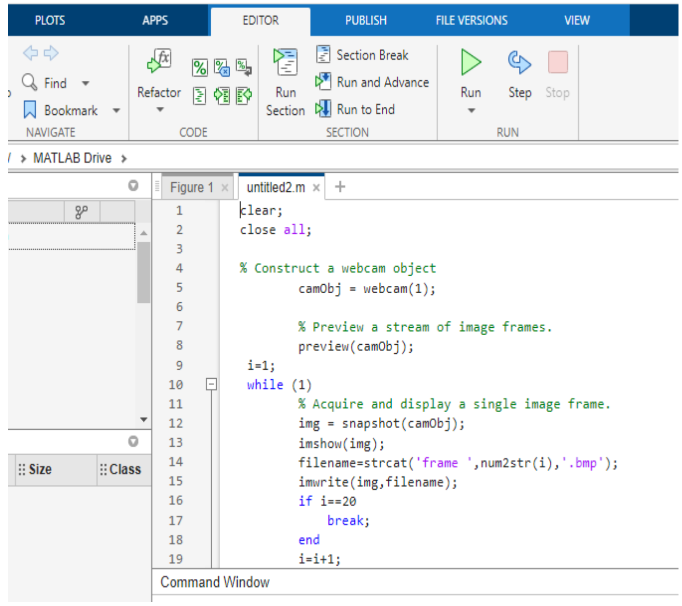


Fig. 4.2.1 Image Snapping

Result:

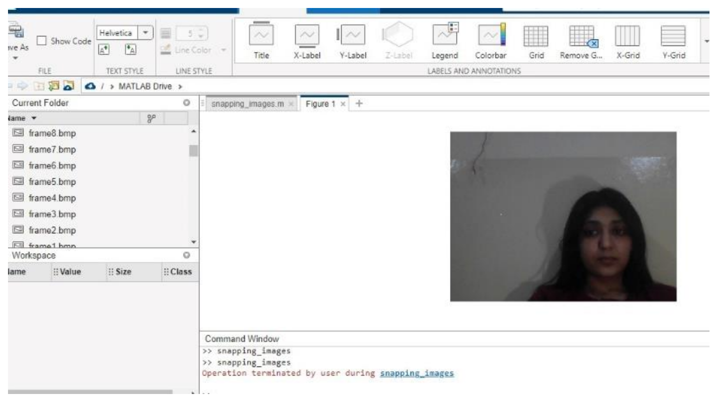


Fig. 4.2.2 Result for Snapping Images from a Webcam

ii. How to detect face through image

Code:

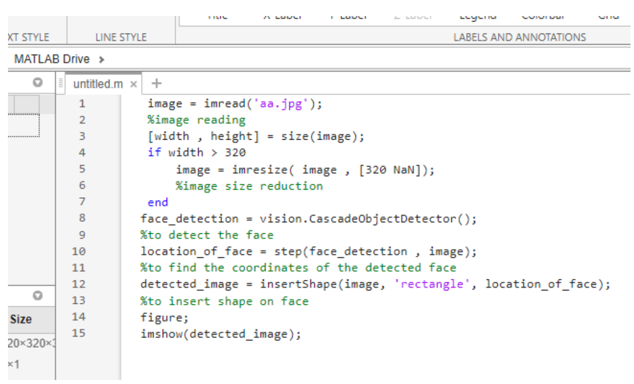


Fig. 4.2.3 Code for Face Detection

Result:

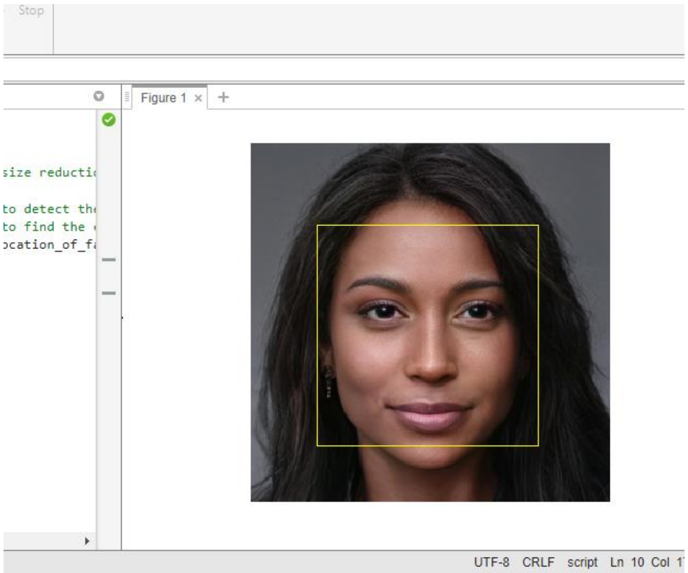
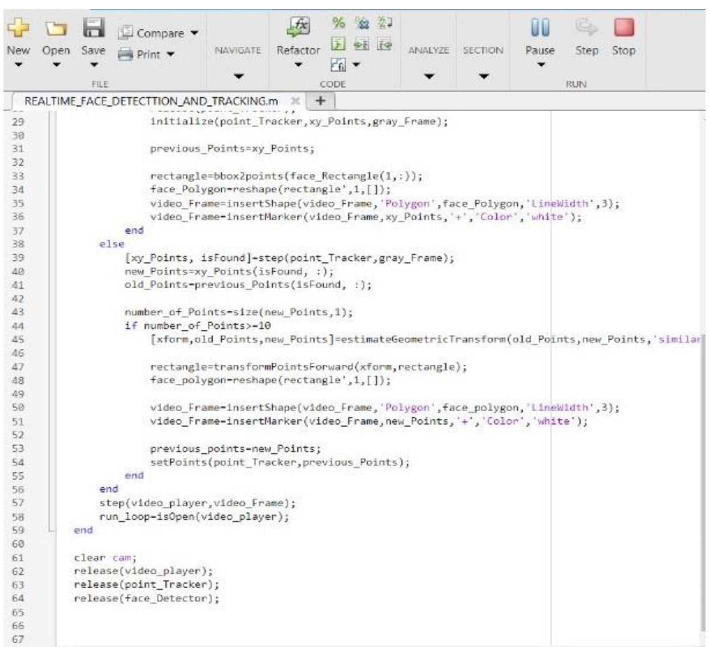


Fig. 4.2.4 Detection of Face through Image

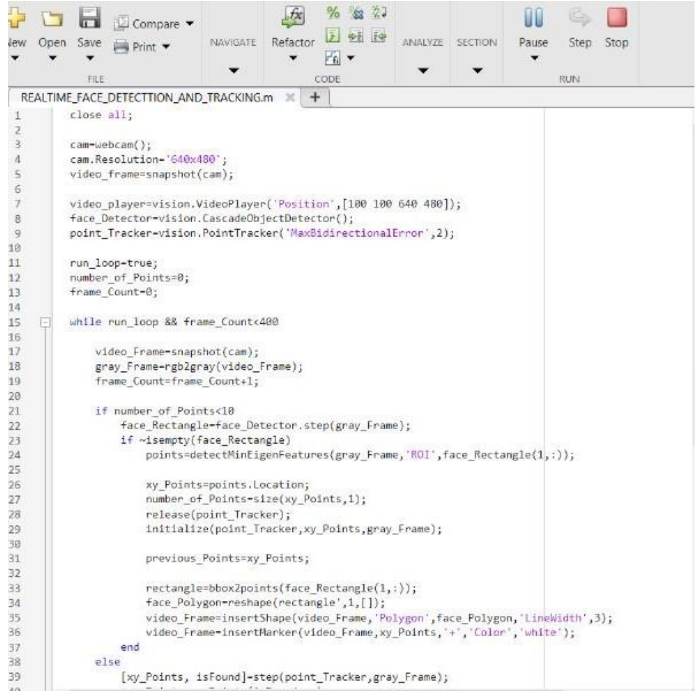
iii. How to detect real-time image

Code:



```
        initialize(point_Tracker,xy_Points,gray_Frame);

        previous_Points=xy_Points;

        rectangle=bbox2points(face_Rectangle(1,:));
        face_Polygon=reshape(rectangle',1,[]);
        video_Frame=insertShape(video_Frame,'Polygon',face_Polygon,'LineWidth',3);
        video_Frame=insertMarker(video_Frame,xy_Points,'+','Color','white');
      end
    else
      [xy_Points, isFound]=step(point_Tracker,gray_Frame);
      new_Points=xy_Points(isFound, :);
      old_Points=previous_Points(isFound, :);

      number_of_Points=size(new_Points,1);
      if number_of_Points>=10
        [xform,old_Points,new_Points]=estimateGeometricTransform(old_Points,new_Points,'similar

        rectangle=transformPointsForward(xform,rectangle);
        face_polygon=reshape(rectangle',1,[]);

        video_Frame=insertShape(video_Frame,'Polygon',face_polygon,'LineWidth',3);
        video_Frame=insertMarker(video_Frame,new_Points,'+','Color','white');

        previous_points=new_Points;
        setPoints(point_Tracker,previous_Points);
      end
    end
    step(video_player,video_Frame);
    run_loop=isOpen(video_player);
end

clear cam;
release(video_player);
release(point_Tracker);
release(face_Detector);
```

Fig 4.2.5 Code for Real-time Face Detection (part 1)



```
close all;

cam=webcam();
cam.Resolution='640x480';
video_frame=snapshot(cam);

video_player=vision.VideoPlayer('Position',[100 100 640 480]);
face_Detector=vision.CascadeObjectDetector();
point_Tracker=vision.PointTracker('MaxBidirectionalError',2);

run_loop=true;
number_of_Points=0;
frame_Count=0;

while run_loop && frame_Count<400

    video_Frame=snapshot(cam);
    gray_Frame=rgb2gray(video_Frame);
    frame_Count=frame_Count+1;

    if number_of_Points<10
        face_Rectangle=face_Detector.step(gray_Frame);
        if ~isempty(face_Rectangle)
            points=detectMinEigenFeatures(gray_Frame,'ROI',face_Rectangle(1,:));

            xy_Points=points.Location;
            number_of_Points=size(xy_Points,1);
            release(point_Tracker);
            initialize(point_Tracker,xy_Points,gray_Frame);

            previous_Points=xy_Points;

            rectangle=bbox2points(face_Rectangle(1,:));
            face_Polygon=reshape(rectangle',1,[]);
            video_Frame=insertShape(video_Frame,'Polygon',face_Polygon,'LineWidth',3);
            video_Frame=insertMarker(video_Frame,xy_Points,'+','Color','white');
        end
    else
        [xy_Points, isFound]=step(point_Tracker,gray_Frame);
```

Fig 4.2.6 Code for Real-time Face Detection (part 2)

Result:

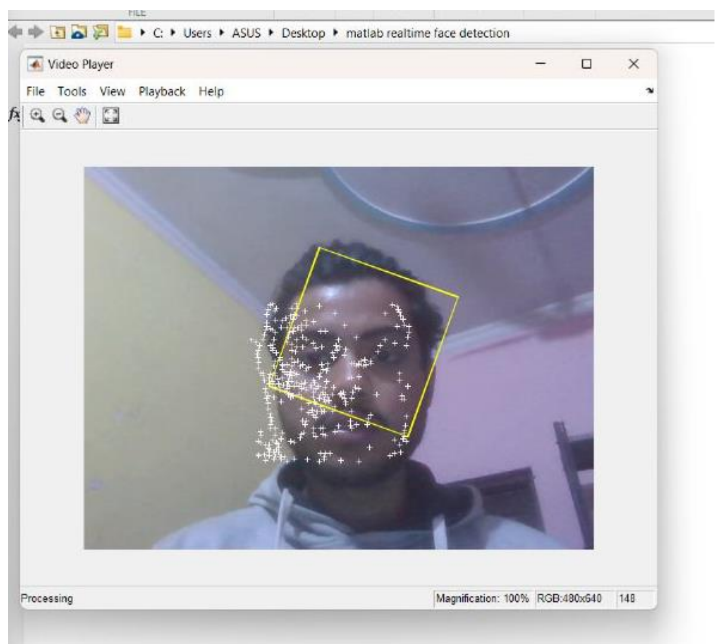


Fig 4.2.7 Result for Real-time Face Detection

iv. Training Model

Code:

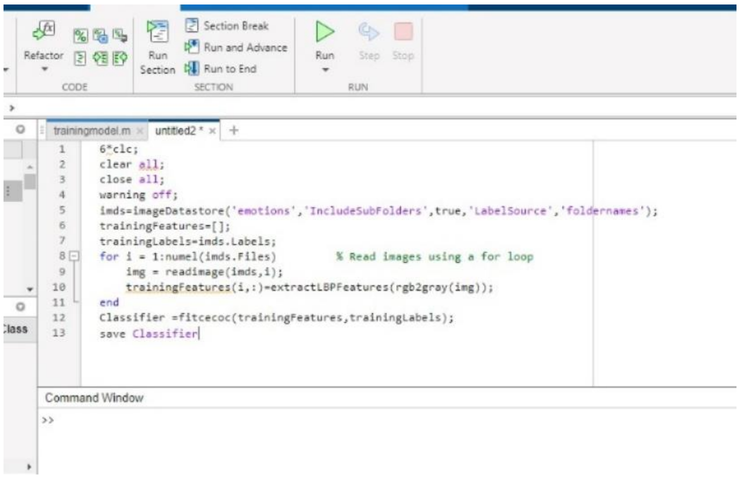
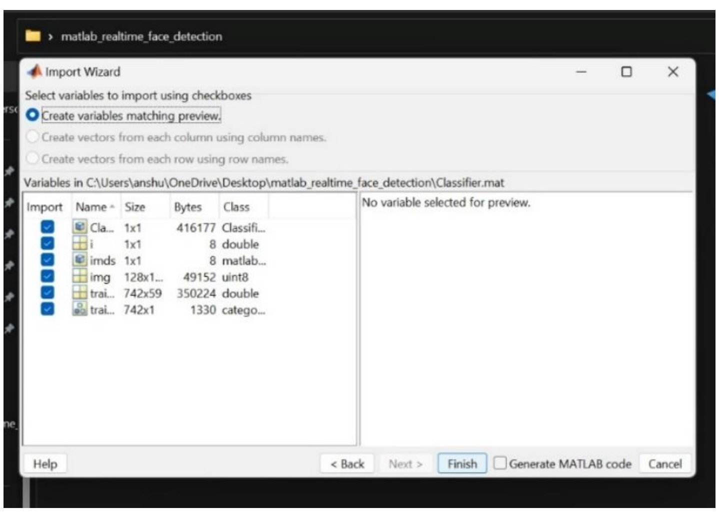


Fig 4.2.8 Code for Training Model

v. Classifier File used for Database Training



Import	Name	Size	Bytes	Class
☑	Cla...	1x1	416177	Classifi...
☑	i	1x1	8	double
☑	imds	1x1	8	matlab...
☑	img	128x1...	49152	uint8
☑	trai...	742x59	350224	double
☑	trai...	742x1	1330	catego...

Fig. 4.2.9 Classifier File for Database Training

5. RESULT

Final Execution:

Code:

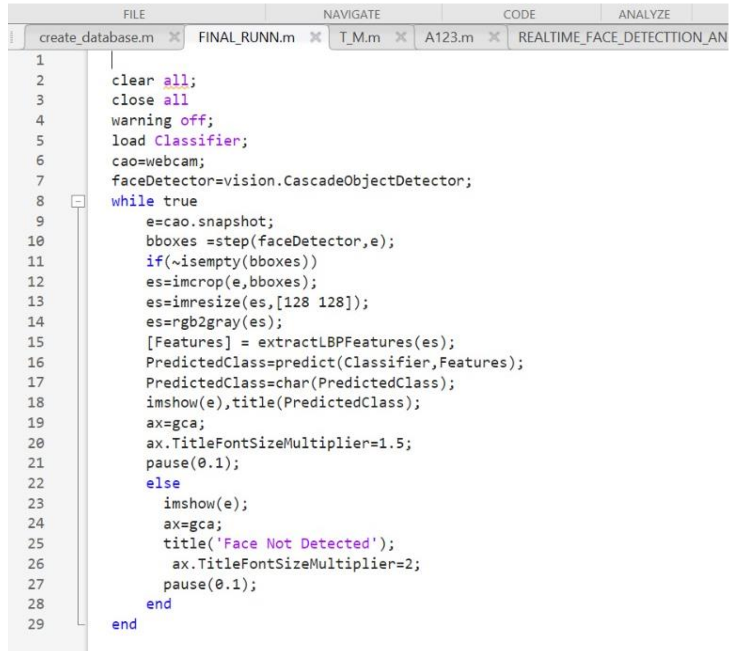


Fig. 5.1 Code for Final Execution

Result:

The emotions detected by our "Real-Time Facial Emotion Recognition" model are shown as below:

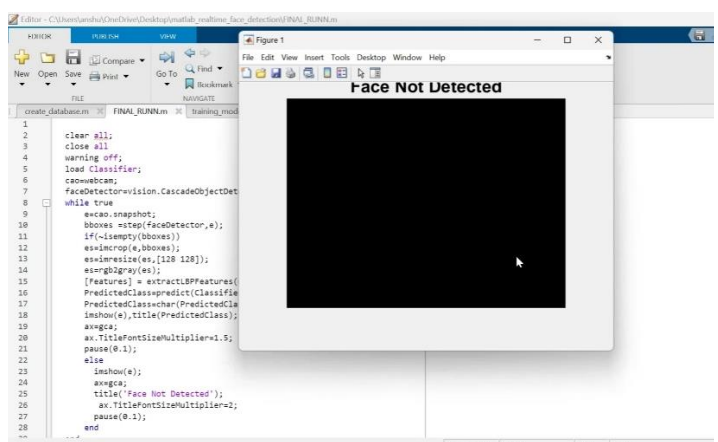


Fig. 5.2 Result when Face not Detected

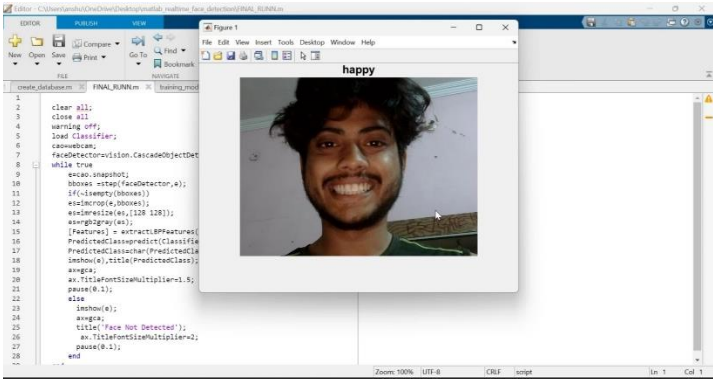


Fig. 5.3 Result when Happy Face Detected

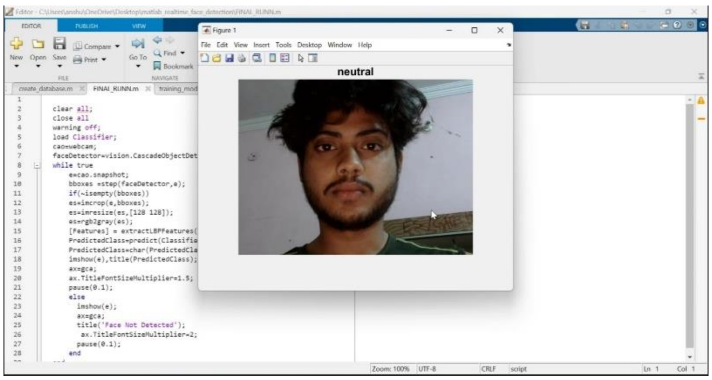


Fig. 5.3 Result when Neutral Face Detected

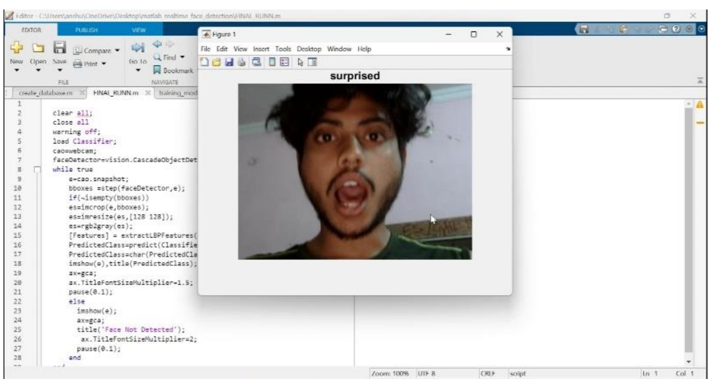


Fig. 5.3 Result when Surprised Face Detected

CONCLUSION

In this project, a method for classifying the facial expressions is proposed. Face identification and facial expression extraction are useful for a number of applications, including robots' vision, security, video surveillance, digital cameras and human-computer interface. This research aimed to develop an enhanced feature extraction and classification system for facial expression recognition using computer vision techniques.

For this study, seven different facial expressions in images of various persons from various datasets were assessed. Happy, sad, neutral, fear, disgust, rage, and surprise are among the emotions that were examined. In this study, pre-processing of the facial expressions of the gathered facial photographs is followed by feature extraction using Local Binary Patterns and facial expression classification using Support Vector Machines based on training datasets of facial images. One occurrence of emotion can be recognized in less than 3 seconds. Any emotion-aware mobile applications can use the system due to its excellent performance and short time requirement. We aim to expand on our work in a subsequent study and take additional emotional input modalities into account.

FUTURE SCOPE

Our system is entirely autonomous and is equipped to handle an image feed. Better face recognition rates are guaranteed by the wide experimental evaluation of the face expressional system. The issue of face classification and the appropriate integration of color and depth information may be further investigated in the future after looking at techniques for addressing expression variance. Future studies can focus on gene alleles that correspond to the geometrical elements of face expressions. The genetic property evolution framework for the face expressional system can be researched to fulfil the requirements of many security models, such as criminal detection, governmental confidential security breaches, etc.

REFERENCES

[1] Nagi, Jawad, Syed Khaleel Ahmed, and Farrukh Nagi. "A MATLAB based face recognition system using image processing and neural networks." In *4th*

International Colloquium on Signal Processing and its Applications, vol. 2, pp. 83-8. 2008.

[2] Gupta, Neha, and Navneet Kaur. "Design and implementation of emotion recognition system by using MATLAB." *International Journal of Engineering Research and Applications (IJERA)* 3, no. 4 (2013): 2002-2006.

[3] Bhadangkar, Dasharath K & Pujari, Jagadeesh & Yakkundimath, Rajesh. (2022). "Identification and Recognition of Facial Expressions Using Image Processing" a survey. 10.13140/RG.2.2.32068.09602.

[4] Chavan, P. M., Manan C. Jadhav, Jinal B. Mashruwala, Aditi K. Nehete, and Pooja A. Panjari. "Real time emotion recognition through facial expressions for desktop devices." *International Journal of Emerging Science and Engineering (IJESE)* 1, no. 7 (2013): 104-108.

[5] Liu, Zhentao, Min Wu, Weihua Cao, Luefeng Chen, Jianping Xu, Ri Zhang, Mengtian Zhou, and Junwei Mao. "A facial expression emotion recognition-based human-robot interaction system." *IEEE/CAA Journal of Automatica Sinica* 4, no. 4 (2017): 668-676.

[6] Murugappan M., Mutawa A., "Facial geometric feature extraction based emotional expression classification using machine learning algorithms", *PLOS ONE*, vol.16, no.2, pp. e0247131, 2021.

[7] A. Kartali, M. Roglić, M. Barjaktarović, M. Đurić-Jovičić and M. M. Janković, "Real-time Algorithms for Facial Emotion Recognition: A Comparison of Different Approaches," *2018 14th Symposium on Neural Networks and Applications (NEUREL)*, Belgrade, Serbia, 2018, pp. 1-4, doi: 10.1109/NEUREL.2018.8587011.

[8] Hauntington, A. H. (2020, August 12). *Humanface*.https://unsplash.com/s/photos/human-face.https://unsplash.com/photos/jzY0KRJopEI

[9] Pahuja, H., Ranjan, P., Ujlayan, A. and Goyal, A., 2022. Convolution Neural Network Based Visual Speech Recognition System for Syllable Identification. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 15(1), pp.139-150.

[10] Pahuja, H., Ranjan, P. and Ujlayan, A., 2017. GPS enabled efficient sound source localisation using microphone array. International Journal of Signal and Imaging Systems Engineering, 10(3), pp.120-135.

[11] Refat, Abu & Singh, Bikash & Rahman, M. (2022). SentiNet: A Nonverbal Facial Sentiment Analysis Using Convolutional Neural Network. International Journal of Pattern Recognition and Artificial Intelligence. 10.1142/S0218001422560079.

[12] Sambare, M. (2020) Fer-2013, Kaggle. Available at: https://www.kaggle.com/datasets/msambare/fer2013/code (Accessed: February 24, 2023).

[13] Parajuli, S., Sunuwar. S., Matang, N., & Shrestha, S. (2017, August 23). A facial expression recognition system a project report. Academia.edu. Retrieved February 24, 2023, from https://www.academia.edu/29632929/A_Facial_Expression_Recognition_System_A_Project_Report

[14] Michel, P., & El Kaliouby, R. (2005). Facial expression recognition using support vector machines. In The 10th International Conference on Human-Computer Interaction, Crete, Greece.

[15] Michel, P., & El Kaliouby, R. (2003, November). Real time facial expression recognition in video using support vector machines. In Proceedings of the 5th international conference on Multimodal interfaces (pp. 258-264). ACM.

[16] *MATLAB LOGIN: Matlab & Simulink*.

MATLAB Login | MATLAB & Simulink.

(n.d.). Retrieved February 27 2023,fromhttps://matlab.mathworks.com/