Facial expression recognition
Group Activity

Kritika Tripathi -21BAC10032
Rakesh Lodhi-21BAC10034
Ashish Barpete-21BAC10037

Name	**Ashish Barpete**
Reg no.	**21BAC10037**
Faculty	**Dr.Anirban Bowmick**
Sub	**Multimedia signal processing**
slot	**C11+12+13+14+15**

Objective: Recognition Facial expression recognition.

Methodology: In this project Facial expression recognition based on Local Binary Patterns & SVM which done in three steps-

1.Database Creation:

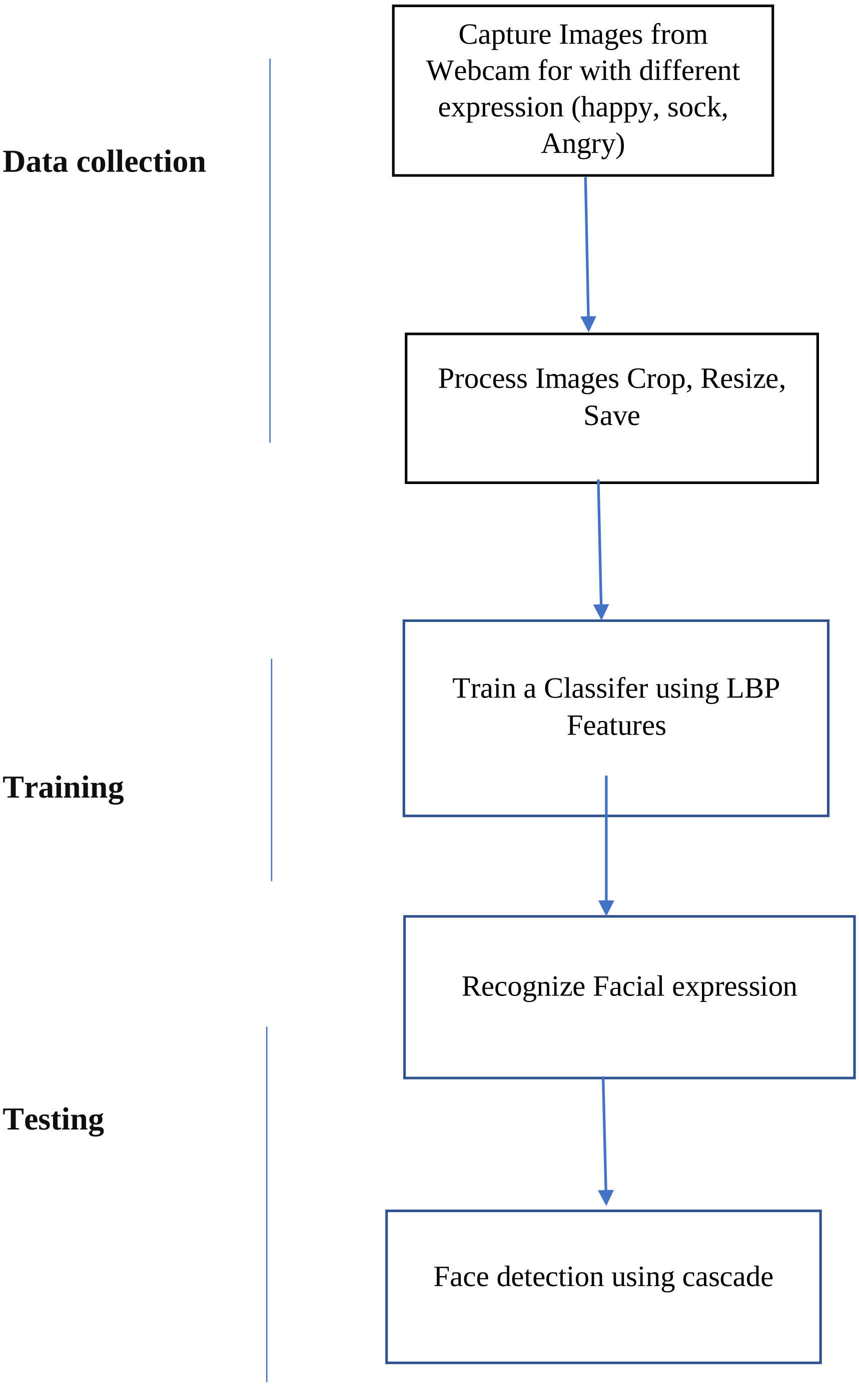
Starts by initializing the webcam (cao) and a face detector (face Detector). It enters a loop where it continuously captures snapshots from the webcam (e). The face detector is then applied to detect any faces in the captured image. If a face is detected, it crops the image around the face region, resizes it to 128x128 pixels, and saves it as a bitmap file with a unique name. The loop continues until a specified number of face images (c) are captured. The captured images will serve as the training data for the face recognition system.

2. Model Training: The code initializes an image data store (imps) that reads the images from the created face database. It creates empty matrices (training Features and training Labels) to store the extracted features and corresponding labels. It then iterates over each image in the image data store, reads the image, extracts Local Binary Pattern (LBP) features from the grayscale version of the image, and stores the features and labels in the matrices. After processing all the images, it trains a multiclass support vector machine (SVM) classifier (Classifier) using the extracted features and labels. Finally, it saves the trained classifier for later use.

3. Testing:

Loads the previously trained classifier (Classifier) and initializes the webcam (cao) and face detector (face Detector). It enters an infinite loop where it continuously captures snapshots from the webcam. The face detector is applied to detect any faces in the captured image. If a face is detected, it crops the image around the face region, resizes it to 128x128 pixels, converts it to grayscale, and extracts LBP features from the image. The trained classifier is then used to predict the class (person) of the face based on the extracted features. The predicted class is displayed as the title of the captured image shown in a figure. If no face is detected, it simply displays the captured image. The loop continues indefinitely, allowing real-time face recognition.

Overall, this code creates a face database by capturing multiple face images, trains a classifier using the extracted features from the database, and then performs real-time face recognition using the trained classifier.


Data collection
Capture Images from Webcam for with different expression (happy, sock, Angry)
Process Images Crop, Resize, Save
Training
Train a Classifer using LBP Features
Recognize Facial expression
Testing
Face detection using cascade

Application :

Emotion-aware User Interfaces: Enhancing user interfaces in various applications such as smartphones, tablets, and computers to adapt and respond based on the user's facial expressions.

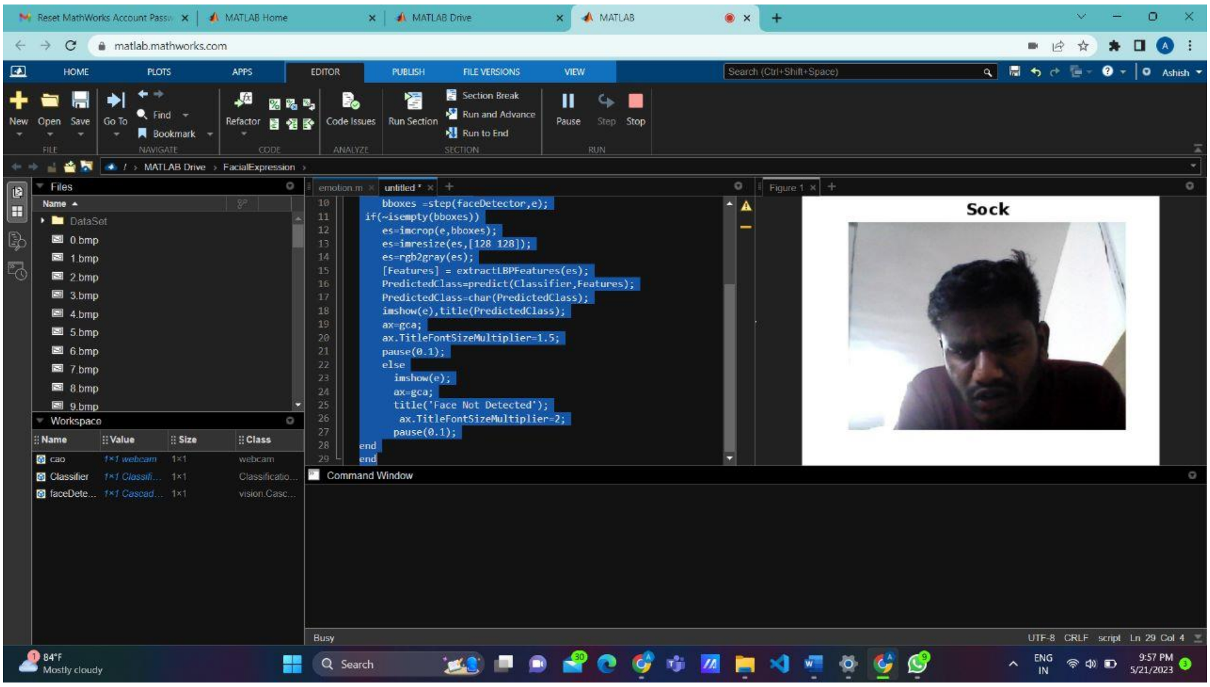
Virtual Meetings and Video Conferencing: Improving remote communication by analyzing participants' facial expressions during virtual meetings and video conferences.

Smart Assistants and Chatbots: Enabling smart assistants and chatbots to recognize and respond to users' emotional states.

Driver Monitoring Systems: Enhancing road safety by integrating facial expression recognition into driver monitoring systems. The technology can detect signs of drowsiness, distraction, or fatigue, providing timely alerts to prevent accidents.

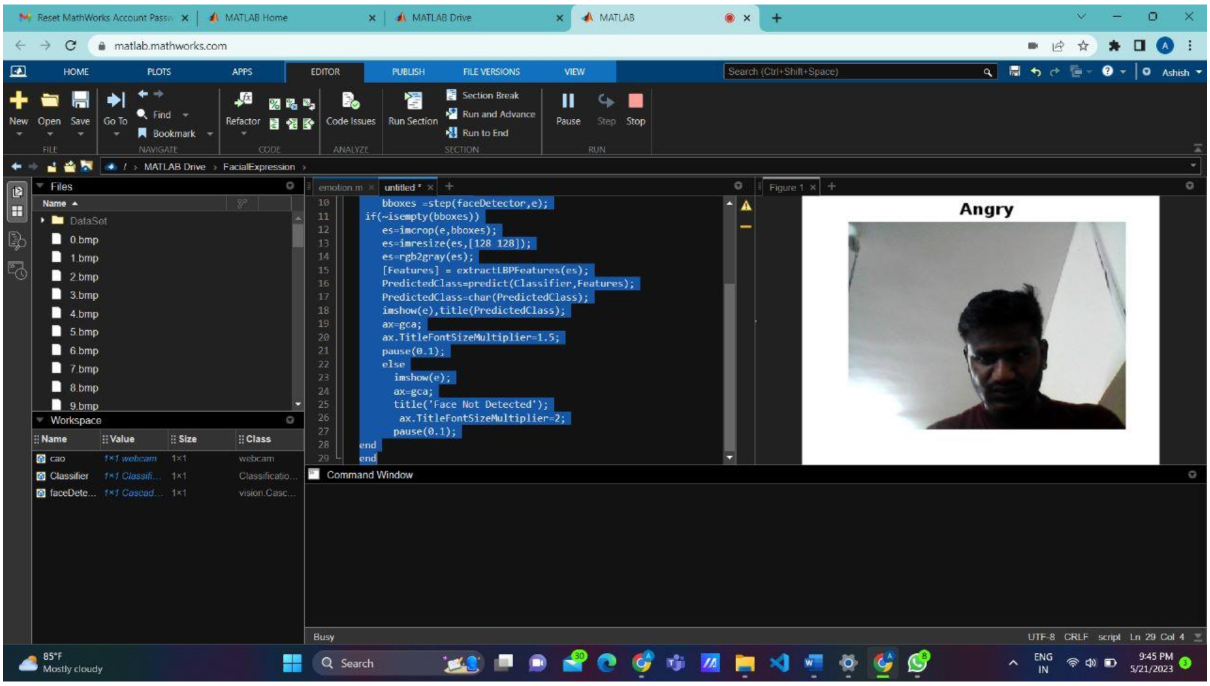
Mental Wellness Applications: Supporting mental wellness by providing emotion analysis tools that individuals can use to track and monitor their emotional states over time.

Output:



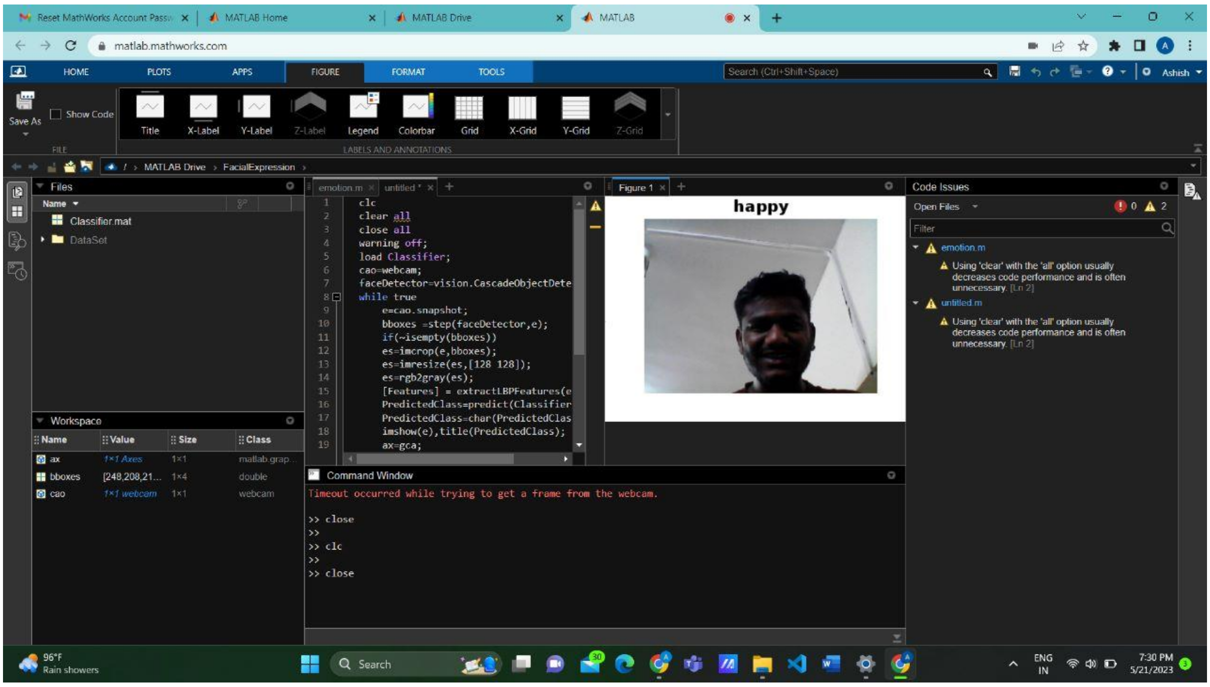
```
    bboxes =step(faceDetector,e);
 if(~isempty(bboxes))
    es=imcrop(e,bboxes);
    es=imresize(es,[128 128]);
    es=rgb2gray(es);
    [Features] = extractLBPFeatures(es);
    PredictedClass=predict(Classifier,Features);
    PredictedClass=char(PredictedClass);
    imshow(e),title(PredictedClass);
    ax=gca;
    ax.TitleFontSizeMultiplier=1.5;
    pause(0.1);
    else
      imshow(e);
      ax=gca;
      title('Face Not Detected');
       ax.TitleFontSizeMultiplier=2;
      pause(0.1);
end
end
```

Name	Value	Size	Class
cao	1×1 webcam	1×1	webcam
Classifier	1×1 Classifi...	1×1	Classificatio...
faceDete...	1×1 Cascad...	1×1	vision.Casc...



Name	Value	Size	Class
cao	1×1 webcam	1×1	webcam
Classifier	1×1 Classifi…	1×1	Classificatio…
faceDete…	1×1 Cascad…	1×1	vision.Casc…

```
    bboxes =step(faceDetector,e);
  if(~isempty(bboxes))
    es=imcrop(e,bboxes);
    es=imresize(es,[128 128]);
    es=rgb2gray(es);
    [Features] = extractLBPFeatures(es);
    PredictedClass=predict(Classifier,Features);
    PredictedClass=char(PredictedClass);
    imshow(e),title(PredictedClass);
    ax=gca;
    ax.TitleFontSizeMultiplier=1.5;
    pause(0.1);
    else
      imshow(e);
      ax=gca;
      title('Face Not Detected');
       ax.TitleFontSizeMultiplier=2;
      pause(0.1);
end
end
```



Name	Value	Size	Class
ax	1×1 Axes	1×1	matlab.grap…
bboxes	[248,208,21…	1×4	double
cao	1×1 webcam	1×1	webcam

```
clc
clear all
close all
warning off;
load Classifier;
cao=webcam;
faceDetector=vision.CascadeObjectDete
while true
    e=cao.snapshot;
    bboxes =step(faceDetector,e);
    if(~isempty(bboxes))
    es=imcrop(e,bboxes);
    es=imresize(es,[128 128]);
    es=rgb2gray(es);
    [Features] = extractLBPFeatures(e
    PredictedClass=predict(Classifier
    PredictedClass=char(PredictedClas
    imshow(e),title(PredictedClass);
    ax=gca;
```

```
Timeout occurred while trying to get a frame from the webcam.

>> close
>>
>> clc
>>
>> close
```

Code Issues

Open Files ⓘ 0 ⚠ 2

emotion.m

⚠ Using 'clear' with the 'all' option usually decreases code performance and is often unnecessary. [Ln 2]

untitled.m

⚠ Using 'clear' with the 'all' option usually decreases code performance and is often unnecessary. [Ln 2]

APPENDEX:

CODE

<u>Code to create Database</u>:

clc

clear all

close all

```
warning off;
cao=webcam;
faceDetector=vision.CascadeObjectDetector;
c=150;
temp=0;
while true
    e=cao.snapshot;
    bboxes =step(faceDetector,e);
    if(sum(sum(bboxes))~=0)
    if(temp>=c)
        break;
    else
    es=imcrop(e,bboxes(1,:));
    es=imresize(es,[128 128]);
    filename=strcat(num2str(temp),'.bmp');
    imwrite(es,filename);
    temp=temp+1;
    imshow(es);
    drawnow;
    end
    else
        imshow(e);
        drawnow;
    end
end
```

Model Training:

```
Model Training:
clc;
clear all;
close all;
warning off;
imds=imageDatastore('Mj','IncludeSubFolders',true,'LabelSource','foldernames');
trainingFeatures=[];
trainingLabels=imds.Labels;
for i = 1:numel(imds.Files)         % Read images using a for loop
    img = readimage(imds,i);
    trainingFeatures(i,:)=extractLBPFeatures(rgb2gray(img));
end
Classifier =fitcecoc(trainingFeatures,trainingLabels);
save Classifier Classifier
```

Testing:

```
Testing:
clc
clear all
close all
warning off;
load Classifier;
cao=webcam;
faceDetector=vision.CascadeObjectDetector;
while true
    e=cao.snapshot;
    bboxes =step(faceDetector,e);
    if(~isempty(bboxes))
    es=imcrop(e,bboxes);
    es=imresize(es,[128 128]);
    es=rgb2gray(es);
    [Features] = extractLBPFeatures(es);
    PredictedClass=predict(Classifier,Features);
    PredictedClass=char(PredictedClass);
    imshow(e),title(PredictedClass);
    ax=gca;
    ax.TitleFontSizeMultiplier=1.5;
    pause(0.1);
    else
      imshow(e);
```

```
   ax=gca;
   title('Face Not Detected');
    ax.TitleFontSizeMultiplier=2;
   pause(0.1);
  end
end
```