PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100 Ft. Road, BSK III Stage, Bengaluru – 560 085

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Title: Image Processing and Data Visualization Using MATLAB		
Course code: UE19CS257B		
Semester: 4th Sem	Branch: CSE	Team Id: 6
SRN: PES1UG19CS375	Name: Rajeshwari R	
SRN: PES1UG19CS439	Name: Sayonika Das	
SRN: PES1UG19CS502	Name: Sravya Yepuri	

PROJECT REPORT

Problem Statement:
To recognise the gestures of hand showing numbers 0 to 5 using AlexNet and building an app on the frontend.

Objectives:
1. Capturing the essence of deep learning and building a simple gesture recognition model using alexNet in matlab.
2. Building a frontend app over the model developed using the App Designer in matlab.

Description:

Data Capture: This module is used to capture images of hand showing 0 (all fingers closed), 1, 2, 3,4 and 5. Each category of image is captured 300 times. Images of hand showing 0 is stored in folder ‘0’, images of hand showing 1 is stored in folder ‘1’ and so on. Images are captured using axes control in GUI and webcam package. Video is continuously shown in axes control. Snapshot of video is taken to capture the images.

Training Data: In this part of the code, we use AlexNet model to train the data. AlexNet is a Convolutional Neural Network which has 8 layers (5 convolutional layers and 3 fully connected layers). 23rd layer in AlexNet architecture (fc8) is assigned a fully connected layer with output size as 7 and 25th layer is the classification layer. Schotastic Gradient Descent with Momentum (sgdm) method is used for optimization for training the data with learning rate as 0.001. MyNet1 file is created and saved with all details of images, layers, network architecture.

Testing Data: Here, we test the trained model by creating a processing area for capturing the gestures. We crop and resize the snapshots as AlexNet requires 227x227 dimension. The model then uses the pre-trained model data in myNet1.mat to predict the captured gesture using the classify function and displays the output.

GUI: Frontend for this project is created using App Designer. Controls used in this project are axes, button, label. Axes control is used to display live video captured by webcam. One button each is added for each module (Data Capture, Train and Test). On clicking these button corresponding modules are called and executed. Label control is used to display title and output.

New Concept Learnt (Explanation):

- **Creating GUI using app designer in MATLAB.**
  App designer is a tool in MATLAB which can be used to create GUI interactively by dragging and dropping the controls. Controls like buttons have function call-backs where the necessary functions are coded into them.

- **Training the model and classifying image data using Transfer Learning with AlexNet.**
  AlexNet is a Deep Convolutional Neural Network which has 8 layers (5 convolutional layers and 3 fully connected layers). Using the fundamental three steps of capturing data (used for training), training the model (using the dataset) and finally testing the model on newer datasets and fine-tuning it to operate with better precision.

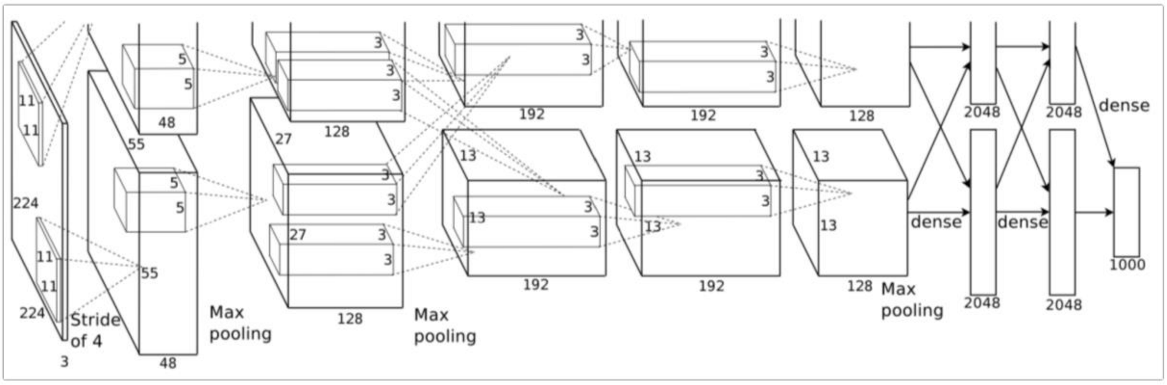


Illustration of AlexNet's architecture. Image credits to Krizhevsky et al., the original authors of the AlexNet paper.

Learning Outcome:

1. Learnt and used many features in MATLAB.
2. Working on this project gave us a good insight into convolution neural networks and Machine Learning.
3. Got to work with the GUI using AppDesigner.
4. Learnt Image and data processing using MATLAB.

Code:

DataCapture.m

```
function DataCapture(wcam, axes)
    x= 0;
    y=0;
    height = 200;
    width=200;
    bboxes = [x y height width];
    temp = 0;
   while temp <= 300
        temp = temp+1;
        e = wcam.snapshot;
        im = image(axes, zeros(size(e), 'uint8'));
        axis(axes, 'image');
        preview(wcam,im);
        filename = strcat("d:\raji\matlab\", num2str(temp));
        filename = strcat(filename,'.bmp');
        es = imcrop(e,bboxes);
        es = imresize(es,[227,227])
        imwrite(es, filename);
        drawnow;
    end
end
```

TrainingData.m

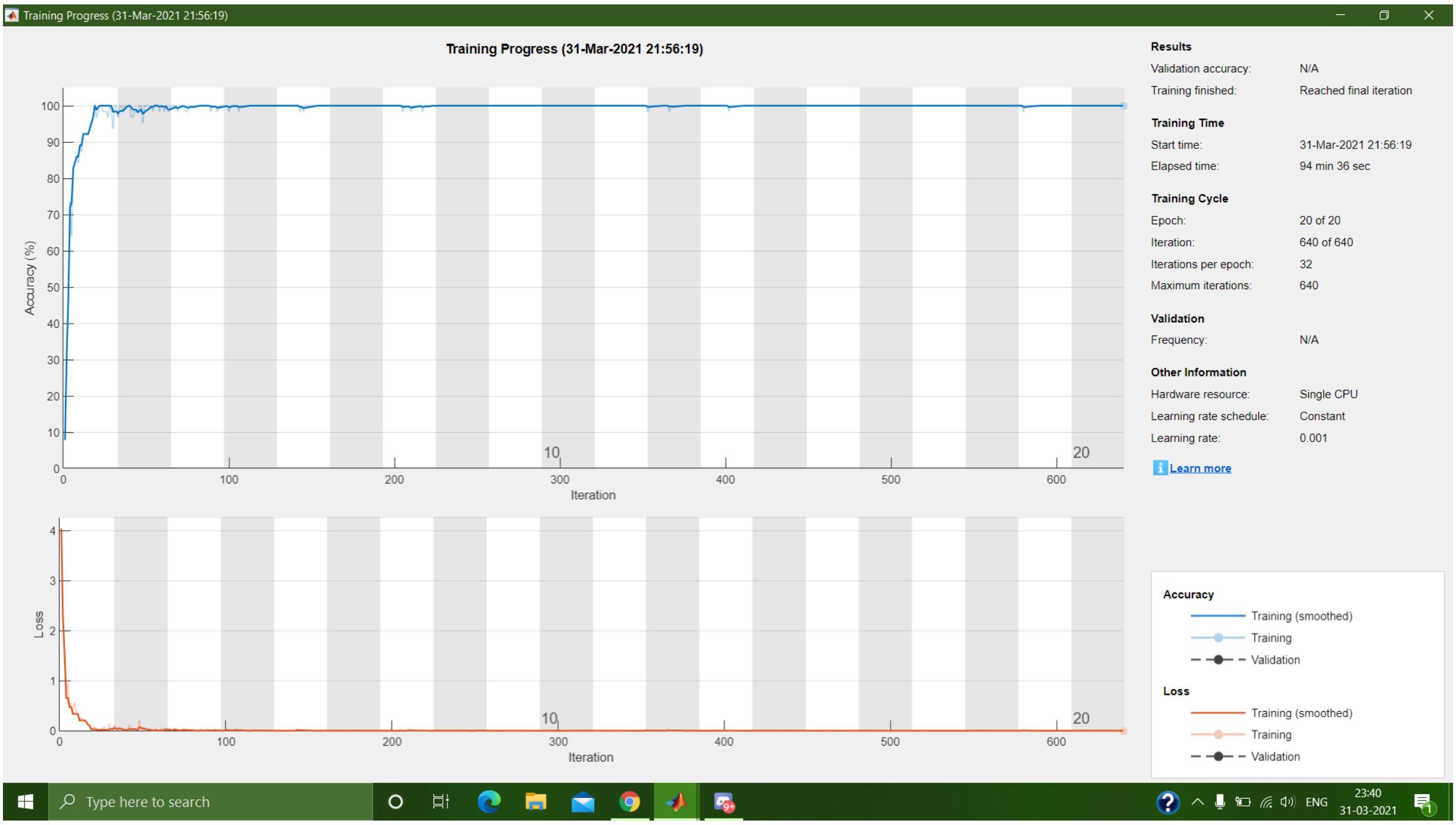
```
function TrainData()
   %AlexNet is a convolutional neural network and has 25 layers
   g = alexnet;
   Layers = g.Layers;
   %layer 23 = fc8 = Fully Connected = 1000 fully connected layer
   %7 = output which should be equal to number of classes (folders =
   blank, 0, 1, 2, 3, 4, 5)
   layers(23) = fullyConnectedLayer(7);
   %layer 25 = Output = Classification Output = Crossentropyex with
   'tench' and 999 other classes
   layers(25) = classificationLayer;
   allImages = imageDatastore('Hand Dataset', 'IncludeSubfolders', true,
   'LabelSource', 'foldernames');
   %sgdm = Schotastic Gradient Descent with Momentum
   %epoch = full training cycle on the entire training data set
   %28 iterations per epoch
   opts=trainingOptions('sgdm','InitialLearnRate',0.001,'MaxEpochs',20,'M
   iniBatchSize',64,'Shuffle','every-epoch','Plots','training-progress');
    myNet1=trainNetwork(allImages, layers, opts);
    save myNet1;
end
```

TestingGesture.m

```
function TestingGesture(wcam, axes, lblOutput)
    load myNet1;
    x=0;
    y=0;
    height=200;
    width=200;
    bboxes=[x y height width];
    while true
        e=wcam.snapshot;
        im = image(axes, zeros(size(e), 'uint8'));
        axis(axes, 'image');
        preview(wcam,im);
        es=imcrop(e,bboxes);
        es=imresize(es,[227 227]);
        label=classify(myNet1,es);
        lblOutput.Text = char(label);
        drawnow;
    end
end
```

Output Screenshots

Training Progress:



Training Progress (31-Mar-2021 21:56:19)

Results

Validation accuracy: N/A

Training finished: Reached final iteration

Training Time

Start time: 31-Mar-2021 21:56:19

Elapsed time: 94 min 36 sec

Training Cycle

Epoch: 20 of 20

Iteration: 640 of 640

Iterations per epoch: 32

Maximum iterations: 640

Validation

Frequency: N/A

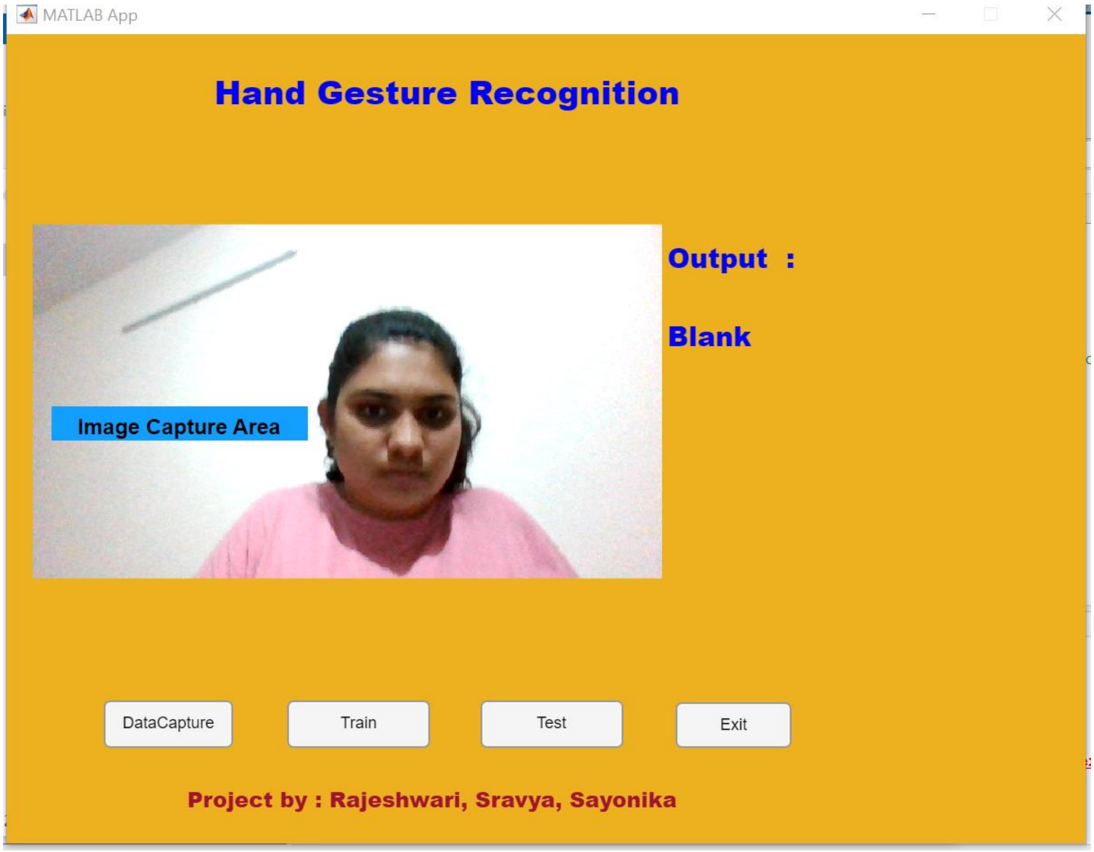
Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

Learning rate: 0.001

blank:



MATLAB App

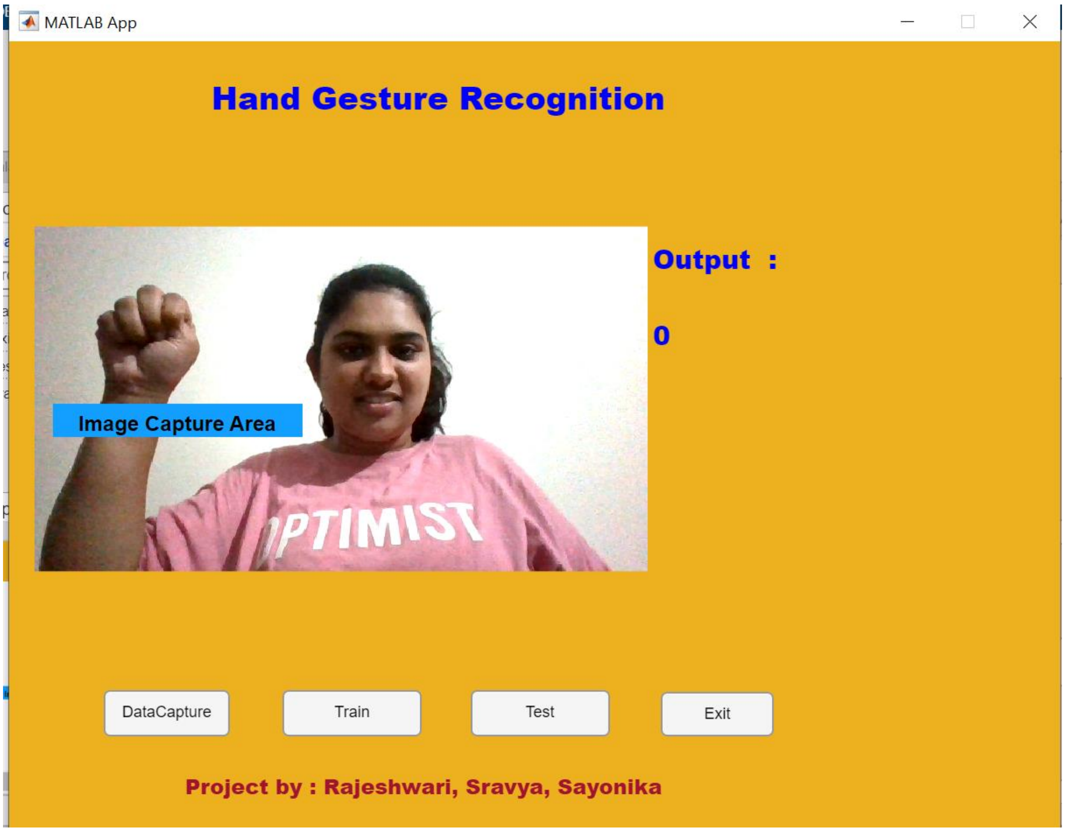
Hand Gesture Recognition

Output :

Blank

Project by : Rajeshwari, Sravya, Sayonika

0:



MATLAB App

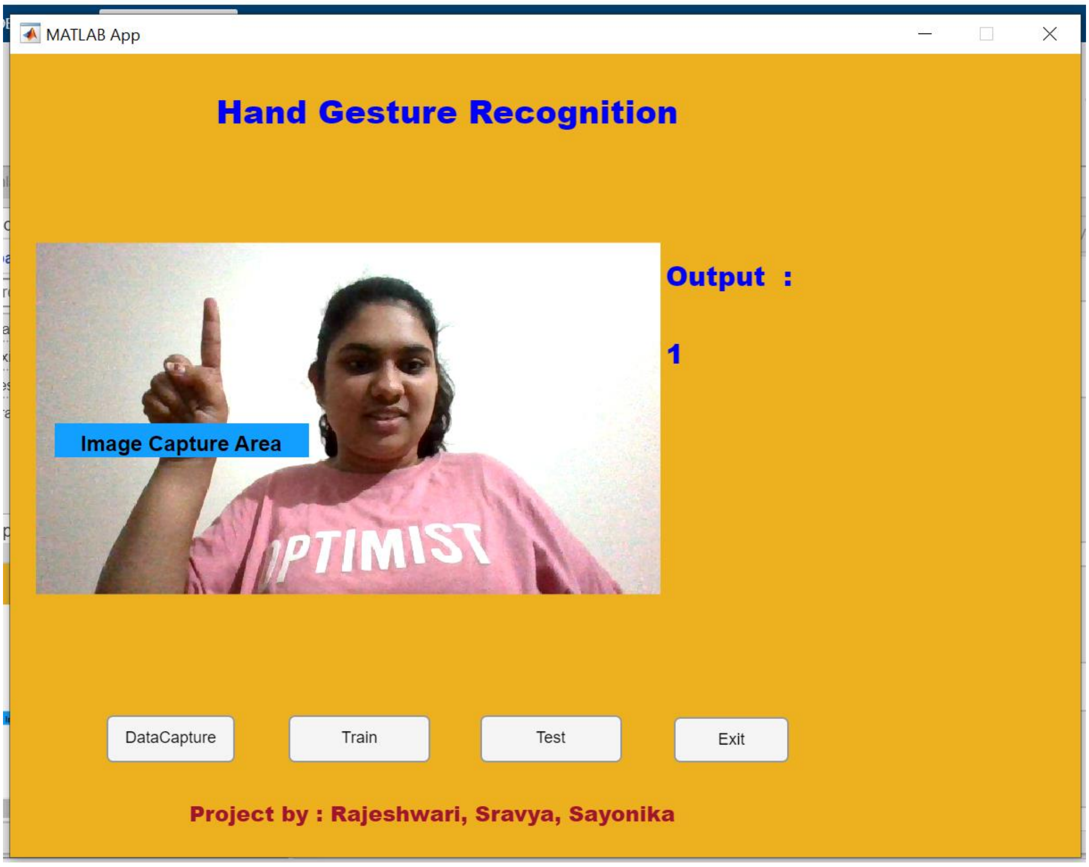
Hand Gesture Recognition

Output :

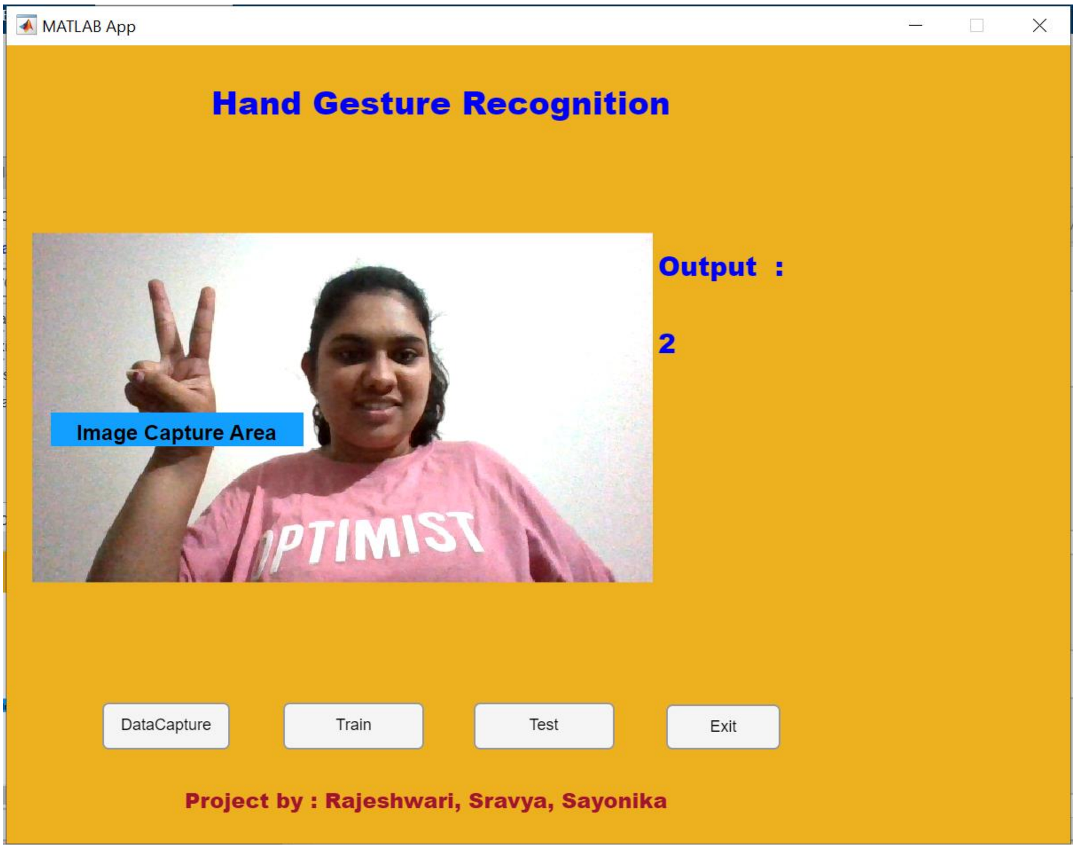
0

Project by : Rajeshwari, Sravya, Sayonika

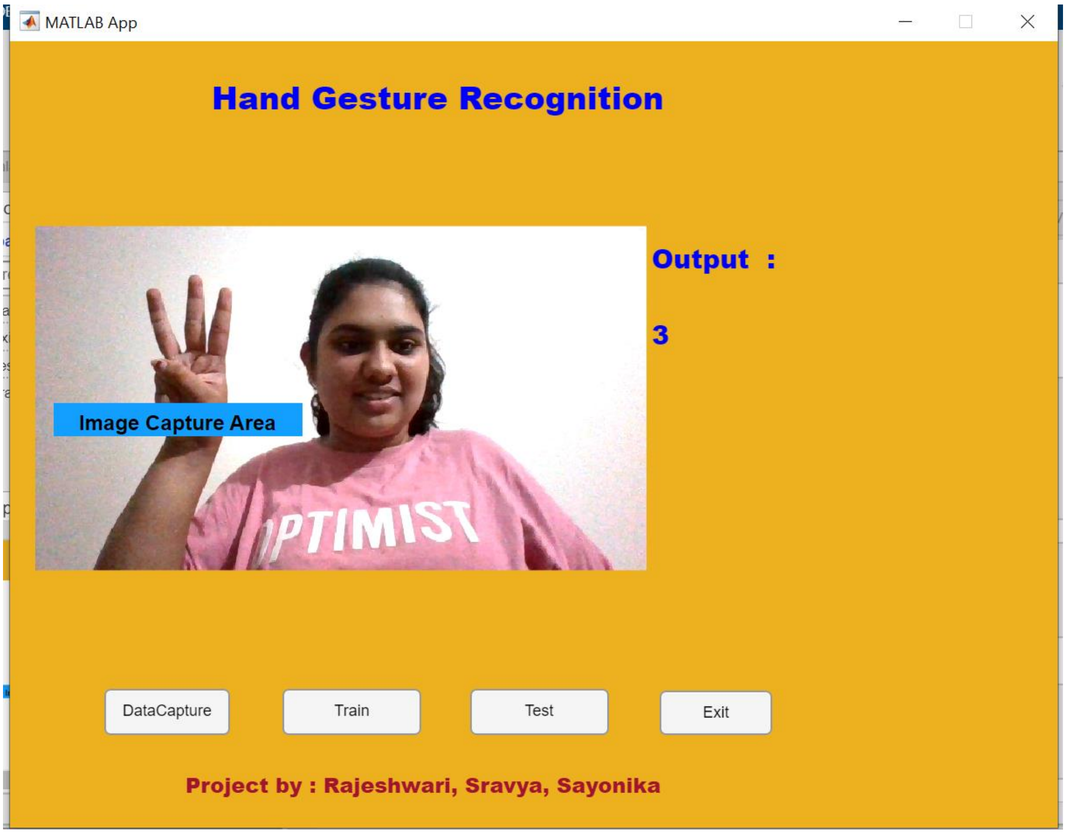
1:


MATLAB App
Hand Gesture Recognition
Output :
1
Image Capture Area
OPTIMIST
DataCapture
Train
Test
Exit
Project by : Rajeshwari, Sravya, Sayonika

2:


MATLAB App
Hand Gesture Recognition
Output :
2
Image Capture Area
OPTIMIST
DataCapture
Train
Test
Exit
Project by : Rajeshwari, Sravya, Sayonika

3:



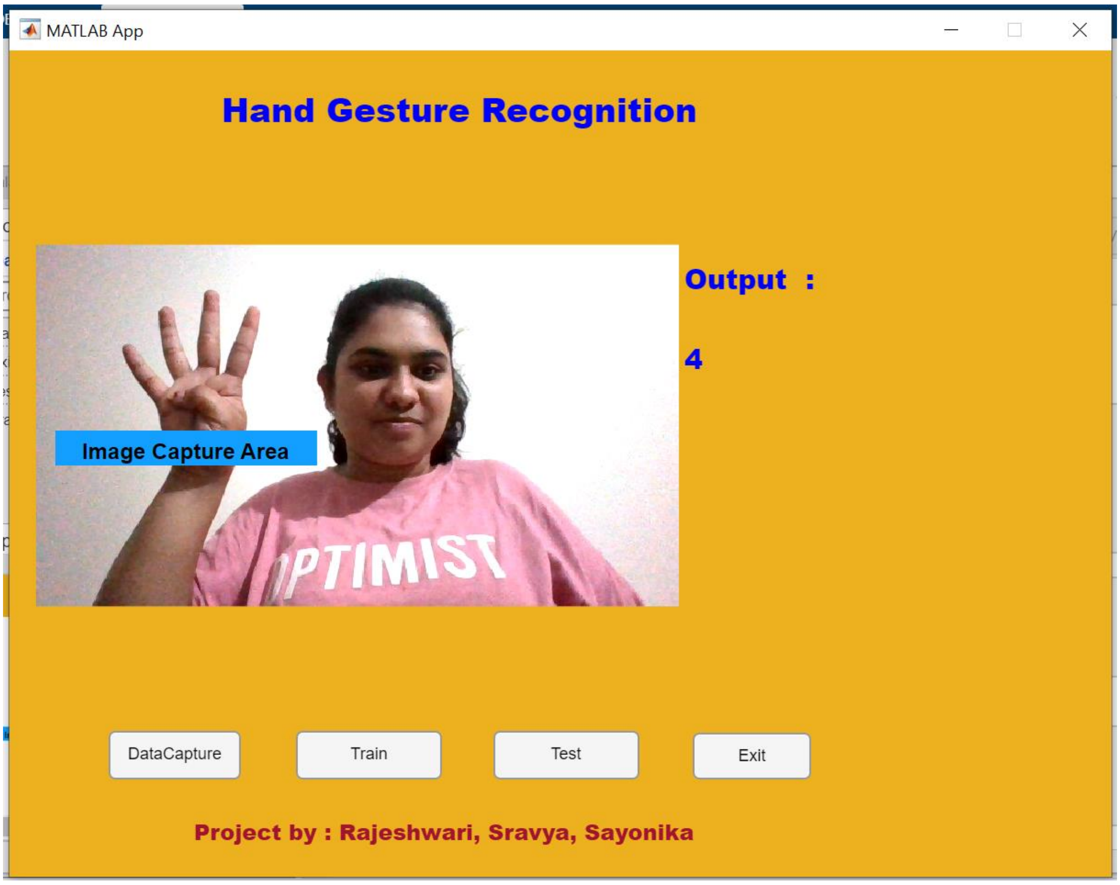
Hand Gesture Recognition

Output :

3

Project by : Rajeshwari, Sravya, Sayonika

4:



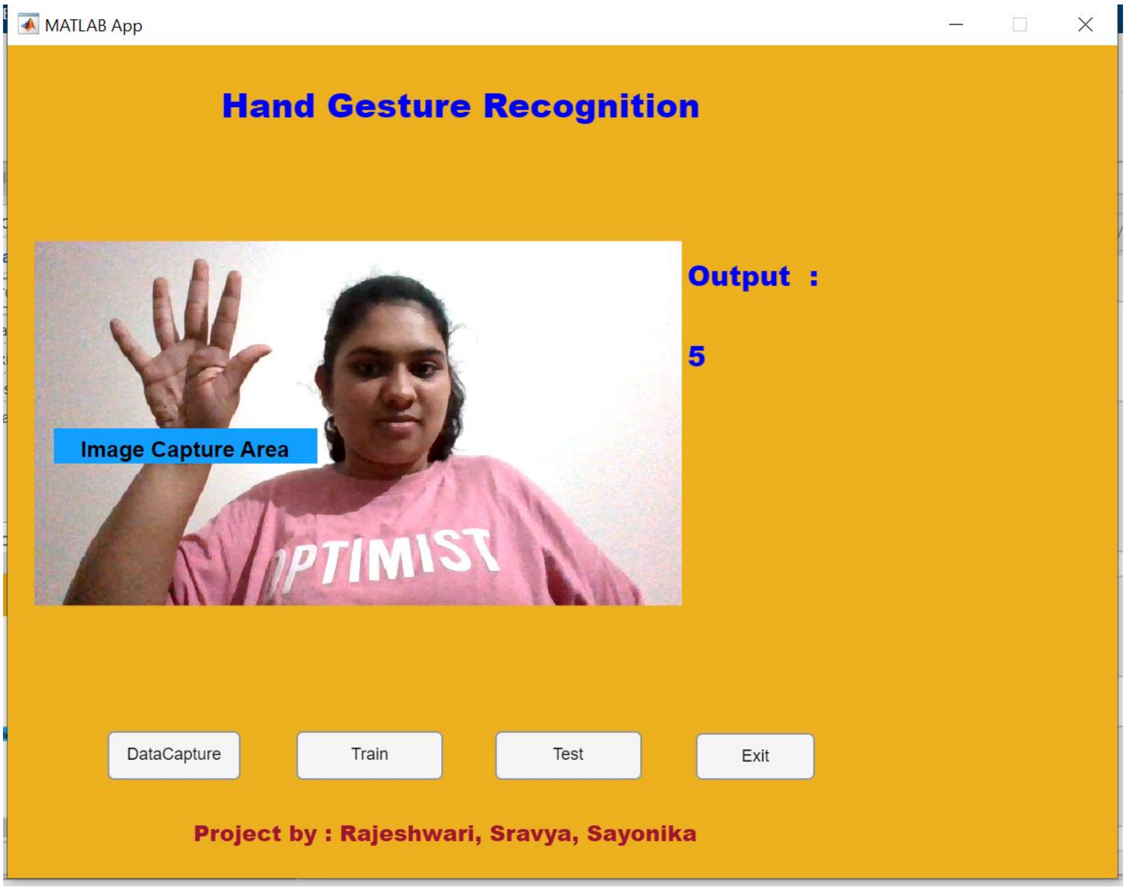
Hand Gesture Recognition

Output :

4

Project by : Rajeshwari, Sravya, Sayonika

5:



Hand Gesture Recognition

Output :

5

Project by : Rajeshwari, Sravya, Sayonika

Name and Signature of the Faculty