Digital Signal Processing
Mini project report

Instructor :- **Dr.Priyanka Kokil**

Team Name :- **Team Alpha**

Project Title :- **Hand Gesture Recognition**

Team Members :-

Ranadeep - EDM18B046
Tulasi - EDM18B047
Bhanuteja - EDM18B049
Varshit - EDM18B050

Date :- 25th June, 2020

1 Abstract

The Healthcare monitoring on a clinical basis involves many implicit communication between the patient and the caretakers.Any misinterpretation leads to adverse effects.
It has always been a challenge for hospitals to eliminate the miscommunications between healthcare providers and those they serve. Whether it is to locate a nurse or responding to a patient's request for more pain medication, ask for any help or call in emergency—it has been outrageously time consuming and difficult to streamline and improve the workflow. Since patients need round-the-clock assistance or sometimes only need help now and then for certain tasks, there should be some solution which uses such technology that can give them peace of mind even when they encounter a situation that they cannot handle independently, or if they have a health issue and need help.

A simple gesture recognition system can precisely interpret the implicit communication to the caretakers or to an automated support device. Simple and obvious hand movements can be used for the above purpose. The hand gesture, during daily life, is a natural communication method mostly used. Gesture recognition is one of the essential techniques to build user-friendly interfaces. The proposed system suggests a novel methodology simpler than the existing ones. This project presents a solution that will not only recognize the hand gestures but will also convert it into speech and text output.

2 Introduction

The demand for caretakers for elderly and the disabled has increased in great ratio. At the same time the population of the elderly in hospitals and care homes has also started increasing. The high cost involved in this expertise makes it always an unachievable target for the healthcare organizations.

Hence in the proposed system, a webcam is used to capture the hand movements of a subject. The aim of this proposal is to enhance the utility of precise communication to the caretaker, including passing information to automated care taking system or a robotic assistance in modern healthcare, by reading the gesture made by the patients or elderly as symbolic. The goal of System Analysis is to completely specify the technical details for the main concept in a concise and unambiguous manner.This system can overcome the limitations of human's aid due to tiredness and lack of timely service.This can also break the barrier of the elderly and disabled towards assistance. Nowadays android and embedded systems are appearing as main prime trends in all applications. Due to that, we change our lifestyle in a smart way. Hand gesture is a method of communication between people.

2.1 Application Domain

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of:

1) Conveying meaningful information or

2) Interacting with the environment.

They constitute one interesting small subspace of possible human motion. A gesture may also be perceived by the environment as a compression technique for the information to be transmitted elsewhere and subsequently reconstructed by the receiver. Hand gestures have boundless applications. Gesture recognition has wide-ranging applications such as the following:

- Developing aids for the hearing impaired, speech impaired.
- Enabling very young children to interact with computers.
- Medically monitoring patients emotional states or stress levels.
- Designing techniques for forensic identification.

3 Brief Literature Survey

In the past, many techniques have been used to convert the hand gesture to text. However, they were limited in terms of their functionalities. Many techniques required gloves with sensors which not only made the application more complex but also expensive. In the other version, the system was limited to a particular background without any noise or disturbance. There were some projects which were heavily dependent on heavy GPUs making it difficult for common man to use the system.
Sign language has different approaches:-

- Kinect sensor : In kinect sensor technique, a sensor is used to detect sign language.
- Image processing : Image processing uses a web camera to capture images.

- Data glove : In data glove technique, flex sensor, accelerometer and motion tracker is used.

- Leap motion : Leap motion controller is sensor. It detects sign language, converts that into computer commands into text and text is converted into speech.

These methods are somewhat complicated and uses advanced devices.

Similar to speech and handwriting, gestures vary between individuals, and even for the same individual between different instances. There have been varied approaches to handle gesture recognition, ranging from mathematical models based on hidden Markov chains to tools or approaches based on soft computing . In addition to the theoretical aspects, any practical implementation of gesture recognition typically requires the use of different imaging and tracking devices or gadgets.These include instrumented gloves, bodysuits, and marker based optical tracking. Traditional 2-D keyboard, pen, and mouse-oriented graphical user interfaces are often not suitable for working in virtual environments. Rather, devices that sense body (e.g., hand, head) position and orientation, direction of gaze, speech and sound, facial expression, galvanic skin response, and other aspects of human behavior or state can be used to model communication between a human and the environment.

4 Methodology

4.1 Gesture Recognition

- The image of hand is captured with a webcam with a dark background and a set of algorithms starts processing it.

- After capturing the image is transformed into gray scale image and again it is converted to black and white image.
- Thereafter we perform a set of morphological operations to remove the surroundings disturbances.

4.1.1 Morphological operations

  - Filling the holes in the image.
  - We are only considered about the shape of the palm therefore we remove all the interior pixels.
  - After we thickens the outline.
  - To be more precise we remove all the isolated pixels in the surroundings

- Note :- Morphological operations are done only for clearing the disturbances in the image these doesn't contribute much part of algorithms.

4.1.2 Algorithm

  - After all these morphological operations we get a clear cut shape and outline of the palm.
  - We pass the information based on the number of fingers raised.Since patients doesn't have much emergency cases we developed algorithm for detection of five gestures.

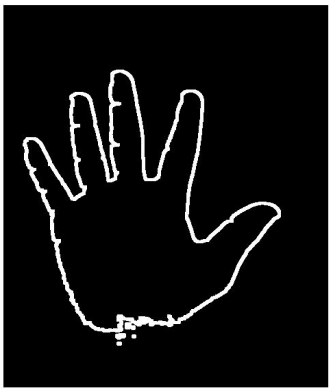
  - From the shape of the image we collect heights of the all pixels on the upper outline.
  - From the heights data we plot the Heights Vs Distance.
  - From the plot we find all the peaks and we fix some threshold height if any of the peak value is greater than threshold then the finger is raised or else it is not raised.
  - Based on the number of fingers raised we display the corresponding information on the screen of the concerned authority.

- To be user friendly We developed an user interface to perform the recognition operation.
- When we switch it ON it starts capturing images and updates the screen for every two seconds based on the user input.We need to switch it ON once and placed near to the patient.

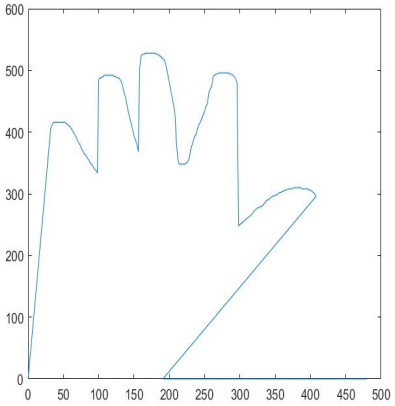
4.2 Algorithm View

Captured image

Black and white image



Edge detected



Outlined image

4.3 Output Type

Figure 1: Different outputs based on user input

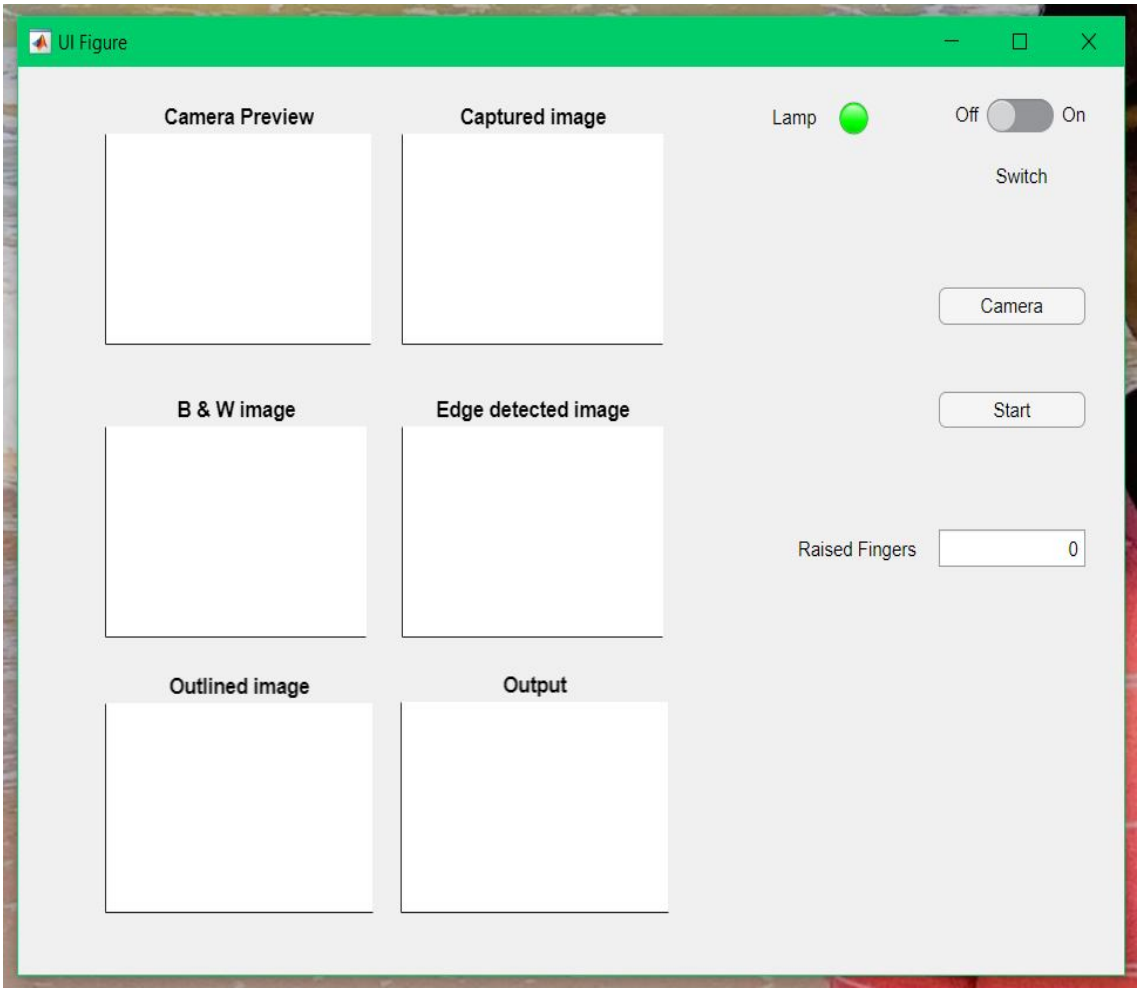


Figure 2: Graphical User Interface Preview

4.4 Challenges on gesture recognition

- **Change in illumination** :-Since we are dealing with black and white images shadows due to uneven illumination of light majorly effects our output.Therefore we are dealing in dark background to get rid of shadows and it makes us to work at low illumination.

- **Rotation Problem** :- We are concluding the outputs based on height it must be vertical therefore we must create our favourable environment.

- **Size Problem** :- Every human being has their different hand and finger sizes we must set the threshold value keeping in mind on every case.

- **Position Problem** :- Position of hand placing in front of webcam is very important.Therefore position of camera before the screen must be placed by studying many different cases.

- **Thumb Recognition** :- Since thumb is opened side wards and different from other fingers.We took much study on it for it's recognition and separate care as taken for it.

5 Work done

- Ranadeep[EDM18B046] :- Algorithm and user interface design.
- Tulasi[EDM18B047] :- Research on problem context.
- Bhanu Teja[EDM18B049] :- Research and study on simulations
- Varshit[EDM18B050] :- Morphological study and handling data base

6 Block Diagram

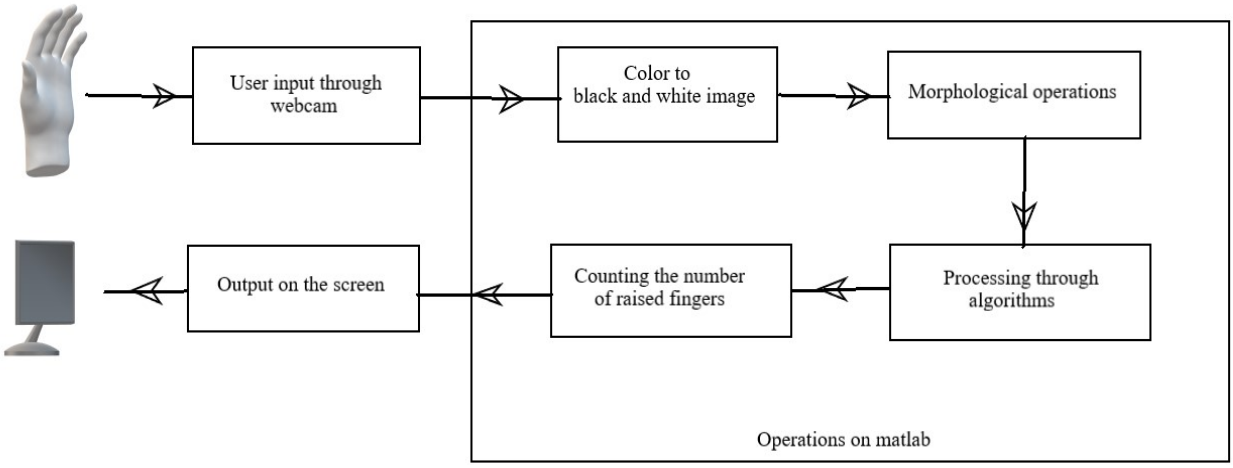


Figure 3: Block Diagram of the system

Initially webcam collects the input from the user and then through the user interface image got read by matlab and then it is converted to gray scale image and from there to black

and white image.This black and white image processed through different morphological operations so clean the palm surroundings and this cleaned image is processed through our designed algorithms to identify the fingers and then counting of fingers takes place from there the corresponding output is displayed on the screen.

7 Execution Procedure

1. Open and switch ON the Matlab application which we had created.

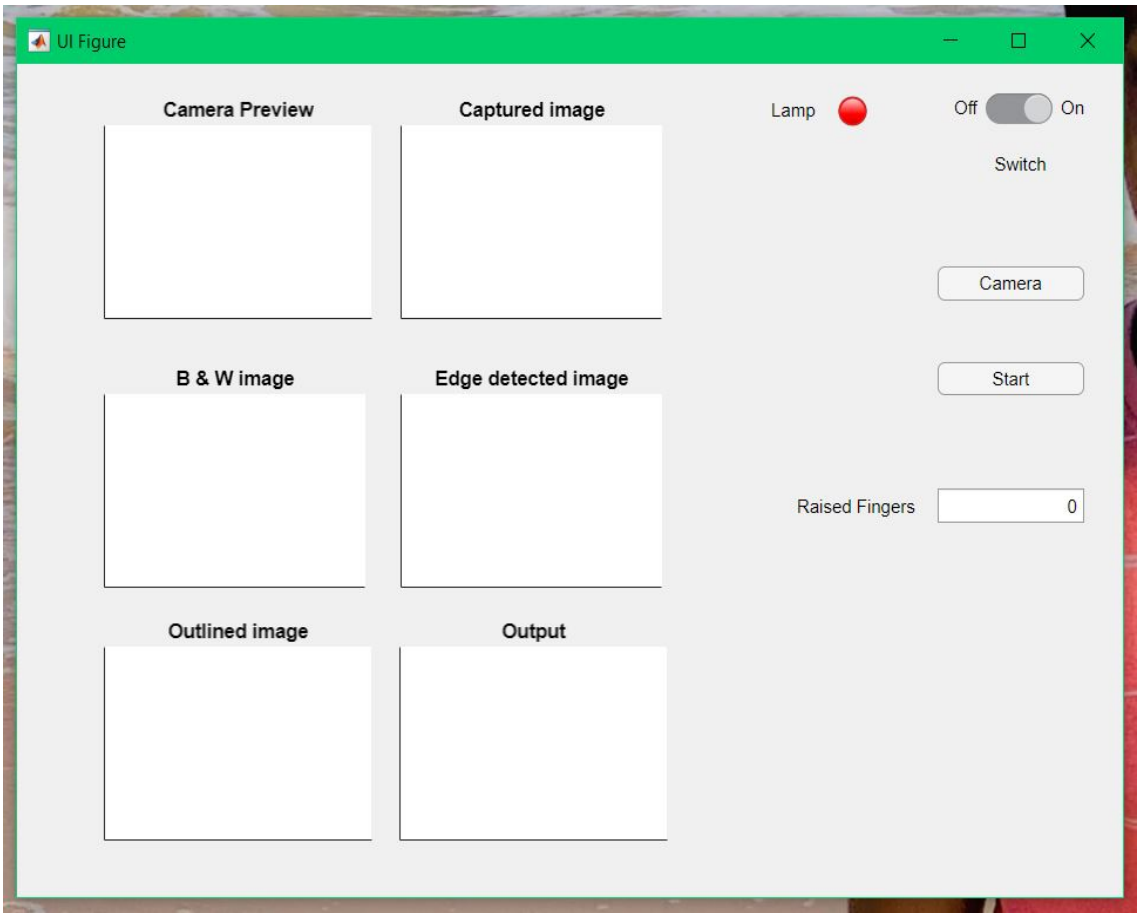


Figure 4: Switched ON application

2. Next tap the camera button it connects the webcam starts preview.

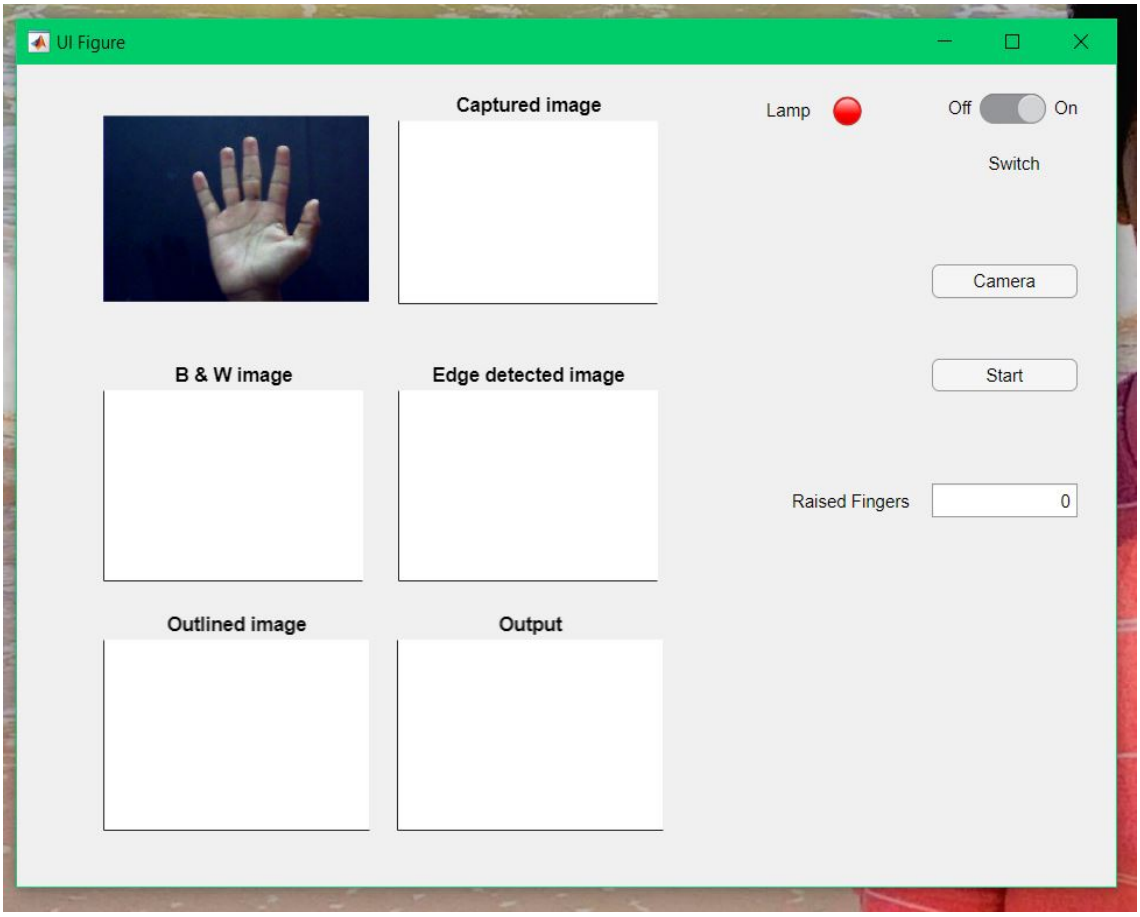


Figure 5: Starting of video preview

3. Next click start button it starts our application and webcam starts capturing for every two seconds and updates the input and all the processes takes place and resultant output is given and the number of raised fingers will be displayed in text box.

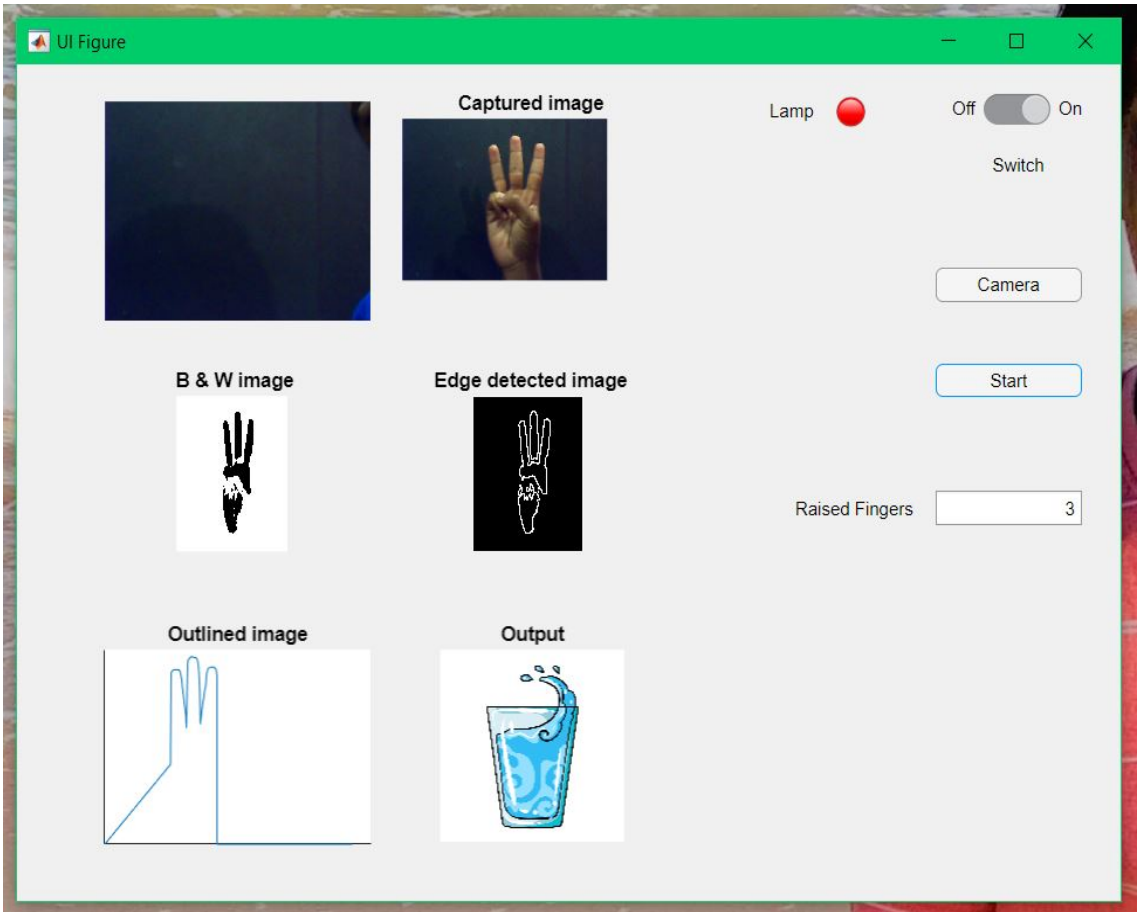


Figure 6: A sample preview after application execution

4. After your usage turn OFF switch it cuts all the connections.

8 Relevant Code

8.1 Hardware

- Components

  1. Personal Computer
  2. Webcam (Normal webcam is enough)

     Specifications of used webcam :-

     - PC compatibility
     - Image Resolution :- 25 Mega Pixels
     - Image Quality :- RGB24 or 1420
     - Frame Rate :- 30fps (MAX)
     - Image Sensor :- High Quality CMOS sensor
     - Image Control :- Color Saturation

8.2 Software

8.2.1 Matlab main function

```
function timef(~,~,app,a)
    b = snapshot(a);                    % captures image
    imshow(b,'parent',app.UIAxes);      % captured image
    d = rgb2gray(b);                    % color to grey scale image
    % Another process for grey to binary
    % e = imadjust(d,[0.77  0.8],[0  1],0.001);
```

```
% e = imresize(e,[640 480]);
lv = graythresh(d);                    % converts grey to binary image
e = imbinarize(d,lv);
e = imresize(e,[640 480]);             % resize the image
e = imcomplement(e);                   % takes the image complement
imshow(e,'parent',app.UIAxes2);        % black and white image
% applying morphological operations
e = imcomplement(e);                   % takes the image complement
f = imfill(e,'holes');                 % fill the holes in the image
f = imcomplement(f);                   % takes the image complement
g = bwmorph(f,'remove');               % remove interior pixels
x = ones(5,5);
h = imdilate(g,x);                     % thickens the image
h = bwmorph(h,'clean';                 % remove isolated pixels
% morphologically processed image
imshow(h,'parent',app.UIAxes3);
s = size(h);                           % getting the size of the image
data = [s(1) zeros(1,s(1)-1)];
xd = zeros(1,s(2));
ind = 2;
% gathering coordinates of upper outline
% Y coordinates in data
% X coordinates in xd
```

```
for i = 10:2:s(2)-10
    for j = 10:2:s(1)-10
        if h(j,i) == 1
            data(ind) = j;
            xd(ind) = i;
            ind = ind + 1;
            break;
        end
    end
end
th = xd(ind-1);
% Adjusting the collected data
for k = 2:s(2)
    if k >= ind
        xd(k) = th;
        th = th +1;
    end
end
for k = 1:s(1)
    if data(k) == 0
        data(k) = s(1);
    end
end
```

```
data = max(data)-data;
data = data(1:s(2));
plot(app.UIAxes4,xd,data);              % plot of outlined image
pks = findpeaks(data);                  % getting peaks from the plot
count = 0;
sp = size(pks);
% counting the number of raised fingers
for i = 1:sp(2)
    if pks(i) > 450
        count = count + 1;
    end
end
% to verify the thumb finger is raised or not
if length(pks) == 5
    if pks(5) > 350 && pks(5) < 420
        count = count + 1;
    end
end
disp(pks);
if length(pks) > 10
    count = 0;
end
app.RaisedFingersEditField.Value = count;
```

```
    % allocate the output w.r.t raised fingers
    if count == 5
        emf = imread('emg.png');
        imshow(emf,'parent',app.UIAxes5);
    elseif count == 4
        med = imread('med.jpg');
        imshow(med,'parent',app.UIAxes5);
    elseif count == 3
        wtr = imread('wtr.jpg');
        imshow(wtr,'parent',app.UIAxes5);
    elseif count == 2
        food = imread('food.jpg');
        imshow(food,'parent',app.UIAxes5);
    elseif count == 1
        com = imread('com.jpg');
        imshow(com,'parent',app.UIAxes5);
    else
        blnk = imread('blank.jpg');
        imshow(blnk,'parent',app.UIAxes5);
    end
end
```

8.2.2 Matlab Application function

```
classdef alpha_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                        matlab.ui.Figure
        SwitchLabel                     matlab.ui.control.Label
        Switch                          matlab.ui.control.Switch
        LampLabel                       matlab.ui.control.Label
        Lamp                            matlab.ui.control.Lamp
        CameraButton                    matlab.ui.control.Button
        StartButton                     matlab.ui.control.Button
        UIAxes                          matlab.ui.control.UIAxes
        UIAxes2                         matlab.ui.control.UIAxes
        UIAxes3                         matlab.ui.control.UIAxes
        UIAxes4                         matlab.ui.control.UIAxes
        UIAxes5                         matlab.ui.control.UIAxes
        RaisedFingersEditFieldLabel     matlab.ui.control.Label
        RaisedFingersEditField          matlab.ui.control.NumericEditField
        UIAxes6                         matlab.ui.control.UIAxes
    end
```

```
properties (Access = private)
    myTimer; % Description
    webcamObject;
    imageObject;
end

methods (Access = private)
    function time(app,a)
        app.myTimer = timer('StartDelay',5,'Period',5,...
        'ExecutionMode','fixedRate','TasksToExecute',3);
        app.myTimer.TimerFcn ={@timef,app,a} ;
        start(app.myTimer);
    end
end


% Callbacks that handle component events
methods (Access = private)

    % Value changed function: Switch
    function SwitchValueChanged(app, ~)
        value = app.Switch.Value;
        if value == "On"
```

```
            app.Lamp.Color = 'Red';
        end
        if value == "Off"
            app.Lamp.Color = 'Green';
            clear global a;
            delete(app.webcamObject);
            delete(app.myTimer);
        end
    end

    % Button pushed function: CameraButton
    function CameraButtonPushed(app, ~)
        global a;
        if app.Switch.Value == "On"
            a=webcam("USB2.0 PC CAMERA");
            a.Resolution='640x480';
            app.webcamObject=a;
            app.imageObject=image(app.UIAxes6);
            axis(app.UIAxes6,'ij');
            res=split(app.webcamObject.Resolution,'x');
            disp(res);
            app.UIAxes6.XLim=[0,str2double(res{1})];
            app.UIAxes6.YLim=[0,str2double(res{2})];
```

```
            app.webcamObject.preview(app.imageObject);
        end
    end

    % Button pushed function: StartButton
    function StartButtonPushed(app, ~)
        global a;
        if app.Switch.Value == "On"
            time(app,a);
        end
    end

    % Callback function
    function RaisedFingersEditFieldValueChanged(~, ~)

    end
end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)
```

```
% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 754 543];
app.UIFigure.Name = 'UI Figure';

% Create SwitchLabel
app.SwitchLabel = uilabel(app.UIFigure);
app.SwitchLabel.HorizontalAlignment = 'center';
app.SwitchLabel.Position = [665 469 41 22];
app.SwitchLabel.Text = 'Switch';

% Create Switch
app.Switch = uiswitch(app.UIFigure, 'slider');
app.Switch.ValueChangedFcn = createCallbackFcn(app,...
@SwitchValueChanged, true);
app.Switch.Position = [662 506 45 20];

% Create LampLabel
app.LampLabel = uilabel(app.UIFigure);
app.LampLabel.HorizontalAlignment = 'right';
app.LampLabel.Position = [511 504 35 22];
app.LampLabel.Text = 'Lamp';
```

```
% Create Lamp
app.Lamp = uilamp(app.UIFigure);
app.Lamp.Position = [561 504 20 20];

% Create CameraButton
app.CameraButton = uibutton(app.UIFigure, 'push');
app.CameraButton.ButtonPushedFcn = createCallbackFcn(app,...
@CameraButtonPushed, true);
app.CameraButton.Position = [629 391 100 22];
app.CameraButton.Text = 'Camera';

% Create StartButton
app.StartButton = uibutton(app.UIFigure, 'push');
app.StartButton.ButtonPushedFcn = createCallbackFcn(app,...
@StartButtonPushed, true);
app.StartButton.Position = [629 329 100 22];
app.StartButton.Text = 'Start';

% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'Captured image')
xlabel(app.UIAxes, '')
```

```
ylabel(app.UIAxes, '')
app.UIAxes.PlotBoxAspectRatio = [1.48175182481752 1 1];
app.UIAxes.XTick = [];
app.UIAxes.YTick = [];
app.UIAxes.Position = [258 374 188 152];

% Create UIAxes2
app.UIAxes2 = uiaxes(app.UIFigure);
title(app.UIAxes2, 'B & W image')
xlabel(app.UIAxes2, '')
ylabel(app.UIAxes2, '')
app.UIAxes2.XTick = [];
app.UIAxes2.YTick = [];
app.UIAxes2.Position = [55 199 188 152];

% Create UIAxes3
app.UIAxes3 = uiaxes(app.UIFigure);
title(app.UIAxes3, 'Edge detected image')
xlabel(app.UIAxes3, '')
ylabel(app.UIAxes3, '')
app.UIAxes3.XTick = [];
app.UIAxes3.YTick = [];
app.UIAxes3.Position = [258 199 188 152];
```

```
% Create UIAxes4
app.UIAxes4 = uiaxes(app.UIFigure);
title(app.UIAxes4, 'Outlined image')
xlabel(app.UIAxes4, '')
ylabel(app.UIAxes4, '')
app.UIAxes4.XTick = [];
app.UIAxes4.YTick = [];
app.UIAxes4.Position = [56 34 191 152];

% Create UIAxes5
app.UIAxes5 = uiaxes(app.UIFigure);
title(app.UIAxes5, 'Output')
xlabel(app.UIAxes5, '')
ylabel(app.UIAxes5, '')
app.UIAxes5.XTick = [];
app.UIAxes5.YTick = [];
app.UIAxes5.Position = [258 34 191 152];

% Create RaisedFingersEditFieldLabel
app.RaisedFingersEditFieldLabel = uilabel(app.UIFigure);
app.RaisedFingersEditFieldLabel.HorizontalAlignment = ...
'right';
```

```
        app.RaisedFingersEditFieldLabel.Position = [528 246 86 22];
        app.RaisedFingersEditFieldLabel.Text = 'Raised Fingers';

        % Create RaisedFingersEditField
        app.RaisedFingersEditField = uieditfield(app.UIFigure,...
        'numeric');
        app.RaisedFingersEditField.Position = [629 246 100 22];

        % Create UIAxes6
        app.UIAxes6 = uiaxes(app.UIFigure);
        title(app.UIAxes6, 'Camera Preview')
        xlabel(app.UIAxes6, '')
        ylabel(app.UIAxes6, '')
        app.UIAxes6.PlotBoxAspectRatio = [1.48571428571429 1 1];
        app.UIAxes6.XTick = [];
        app.UIAxes6.XTickLabel = '';
        app.UIAxes6.YTick = [];
        app.UIAxes6.YTickLabel = '';
        app.UIAxes6.Position = [55 362 192 176];

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
```

```
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = alpha_exported

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        if nargout == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
```

```
            delete(app.UIFigure)
        end
    end
end
```

8.2.3 Some inbuilt matlab functions used

- **snapshot** :- Captures image from video preview.
- **rgb2gray** :- Converts color image to gray scale image.
- **imbinarize** :- Converts gray scale image to binary image.
- **imresize** :- resize the image to required size.
- **imcomplement** :- inverts the image.
- **imshow** :- displays the given image.
- **imfill with holes operation** :- fill in single pixel ”holes”(background pixels surrounded by foreground pixels).
- **bwmorph with remove operation**:- Remove interior pixels.
- **bwmorph with clean operation** :- Remove isolated foreground pixels.
- **imdilate** :- perform dilation on image (dilation means thickens or growth of objects in the image).
- **findpeaks** :- finds the maximas from the data.
- **length** :- find length of the matrix.

9 Hardware Setup

Figure 7: Setup
1.Personal computer 2.Webcam 3.Dark background

10 Simulation Results and Discussions

- Initially all outputs has been observed and studied in resolution 640 x 480.
- From there by observing various gestures and various fingers height when they are raised and when they are not raised we came to conclusion on threshold height.

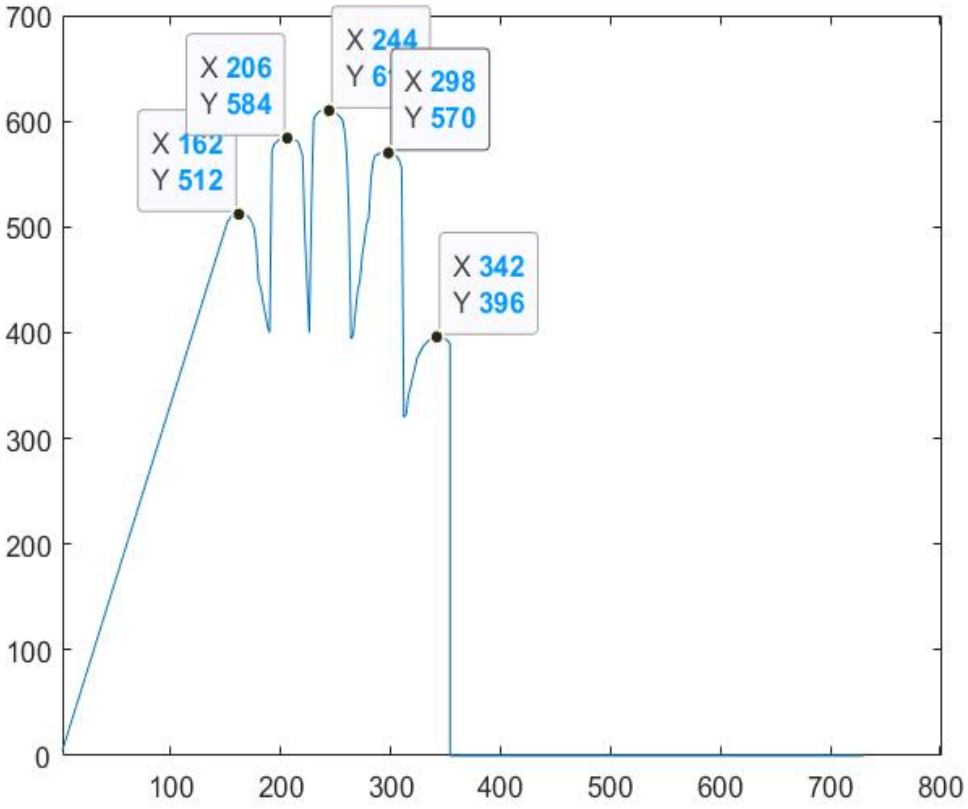


Figure 8: Simulation study sample

- From this study we assigned threshold height value for thumb and other fingers.

- Threshold values :-
  - Others fingers except thumb :- >450
  - For thumb :- >350 and <420
- Outputs

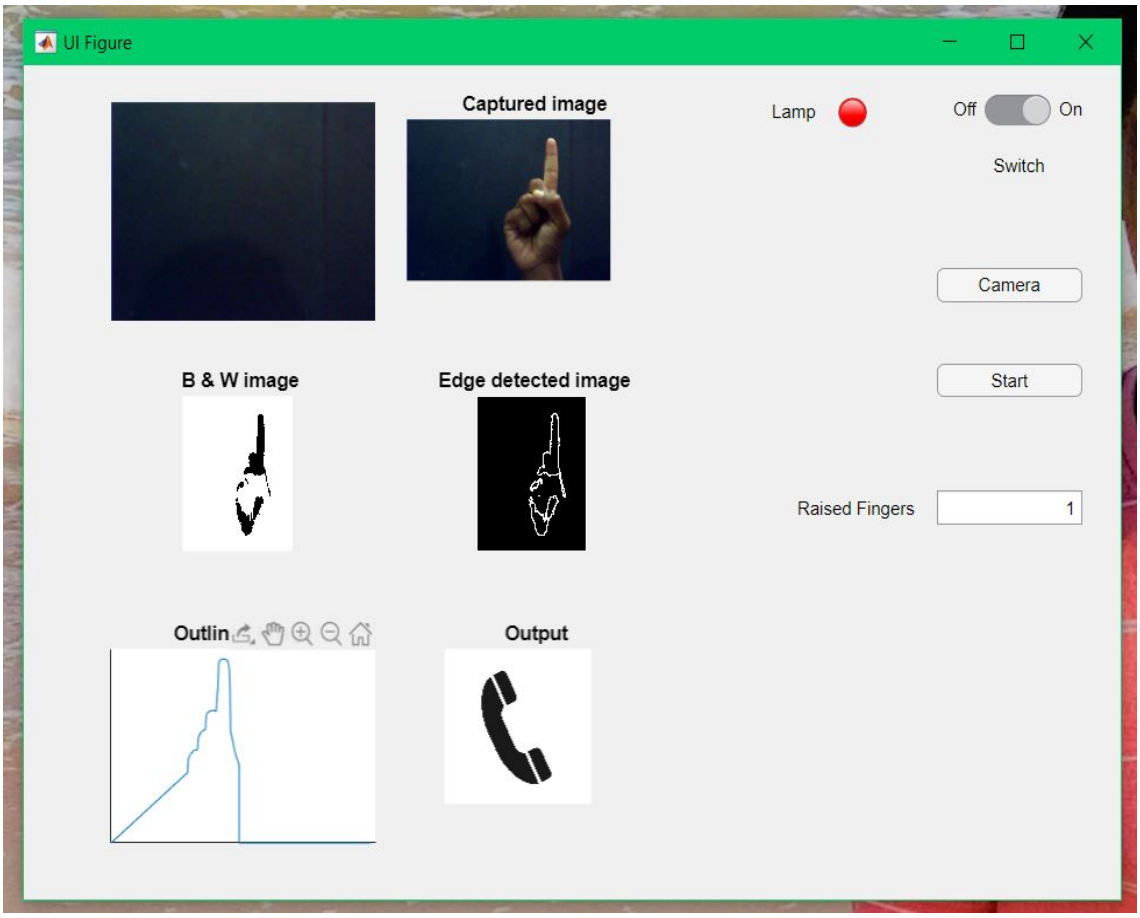


Figure 9: One finger raised

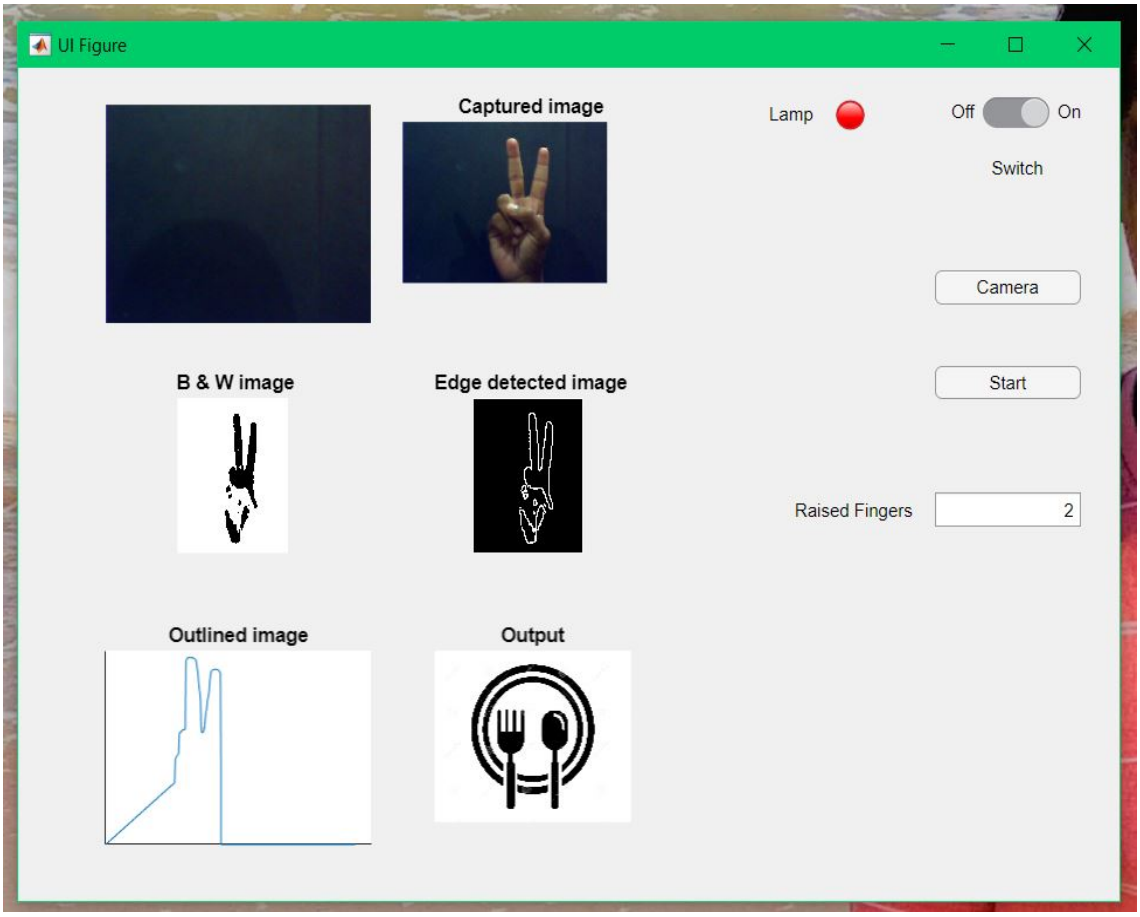


Figure 10: Two fingers raised

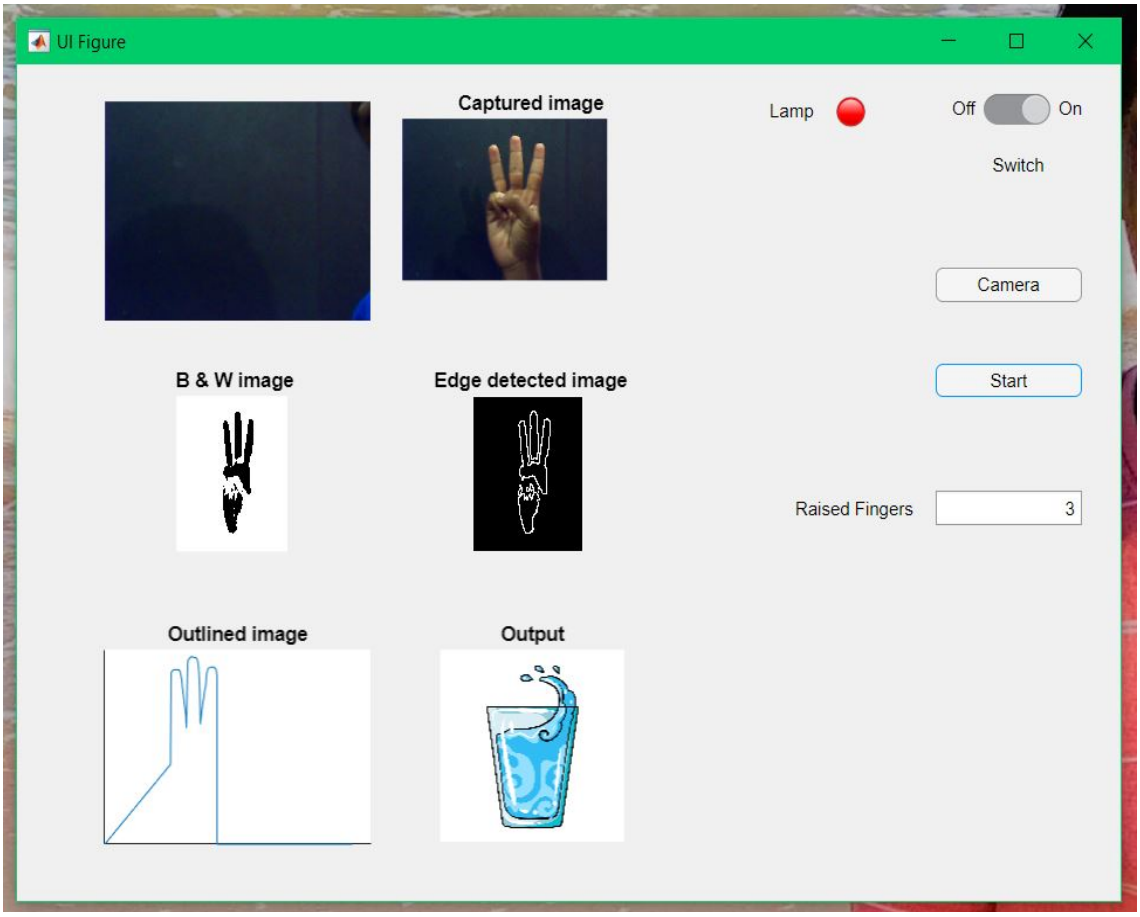


Figure 11: Three fingers raised

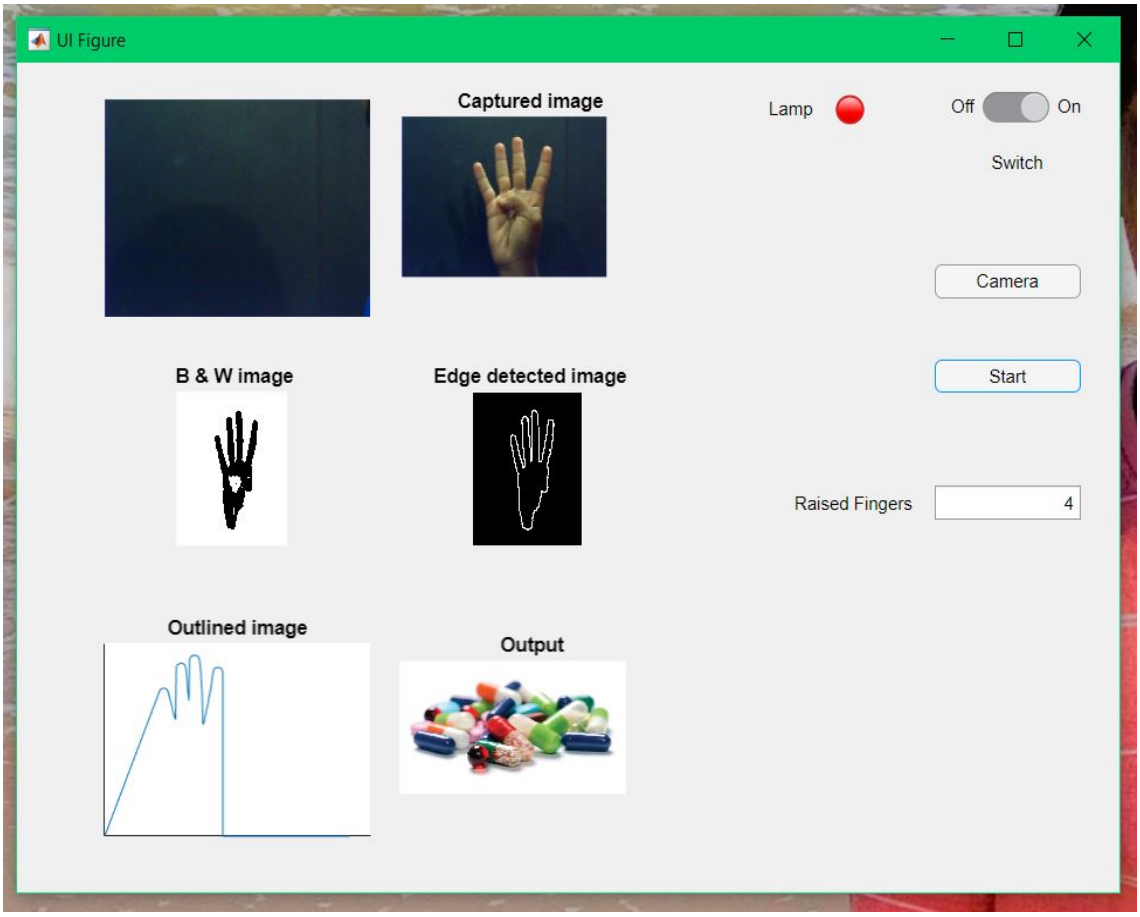


Figure 12: Four fingers raised

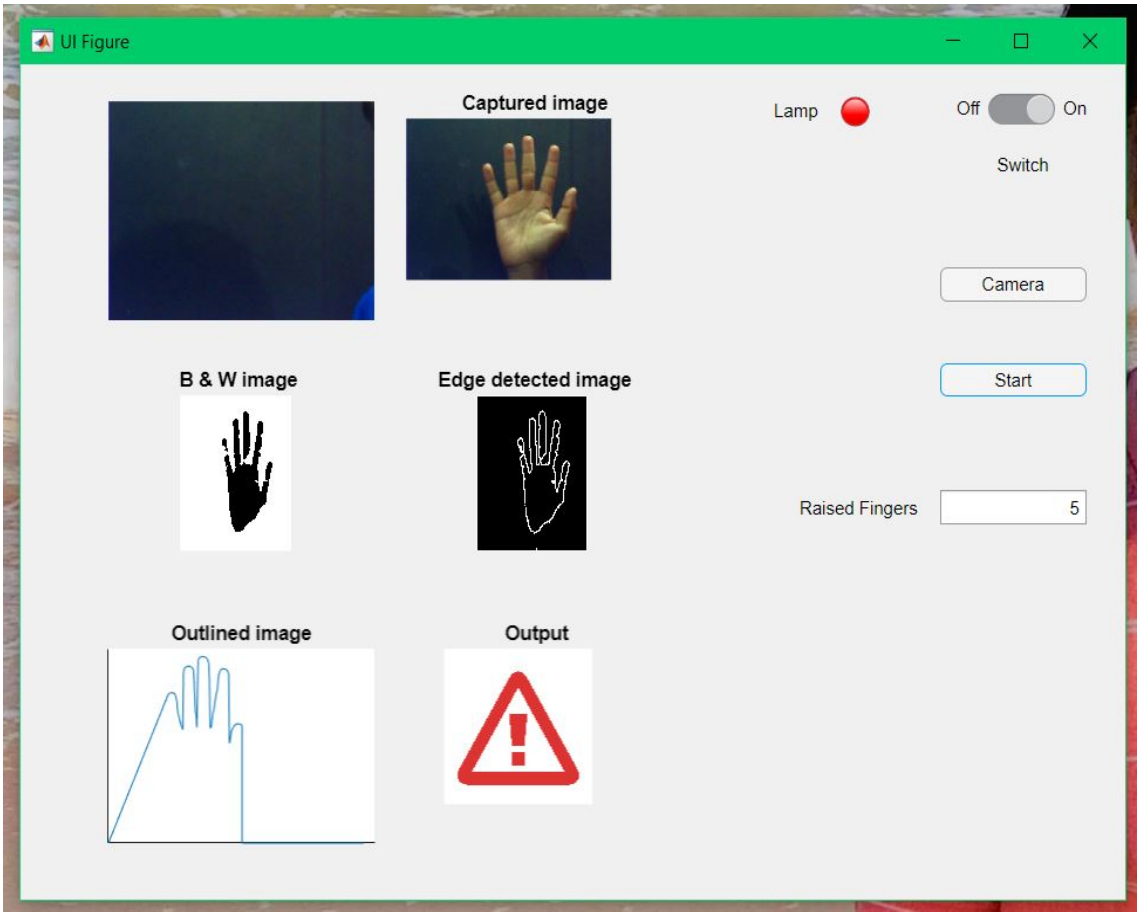


Figure 13: Five fingers raised

- In real life situation the output in our application is displayed on the screen of the concerned authorities in the hospitals and also where ever the project is used.

11 Conclusion

At present, artificial neural networks are emerging as the technology of choice for many applications, such as pattern recognition, gesture recognition, prediction, system identification and control.The application can be integrated with other mobile and IoT devices to improve user interaction and make the system more robust. The accuracy of the program can be further improvised by using neural networks. An alternate stress could be put on the use of the application in the fields of medicines, military, governance etc. A genuine blend of various technologies in mentioned fields could make way for power tools and applications which will serve the community around the world. Finally, the use can be further designed to make it more accessible to the consumers. The whole point of making the solution as a commercially viable product for the users is to help the community around the world.

References

[1] Rafael C. Gonzalez,Richard E. Woods, and Steven L. Eddins *Digital Image Processing Using MATLAB*. McGraw-Hill Education (India).