EE 304 - Design Project

Group No.	34		
Name	P. Harsha Vardhan	**Name**	M. Lalitesh
Roll Number	11010281	**Roll Number**	11010821
Supervisor(s)	Dr. Prithwijit Guha		
Project Title	Real Time face gesture recognition (two face gestures) and mimicking on Face prototype		

The Proposed Application: An Introduction

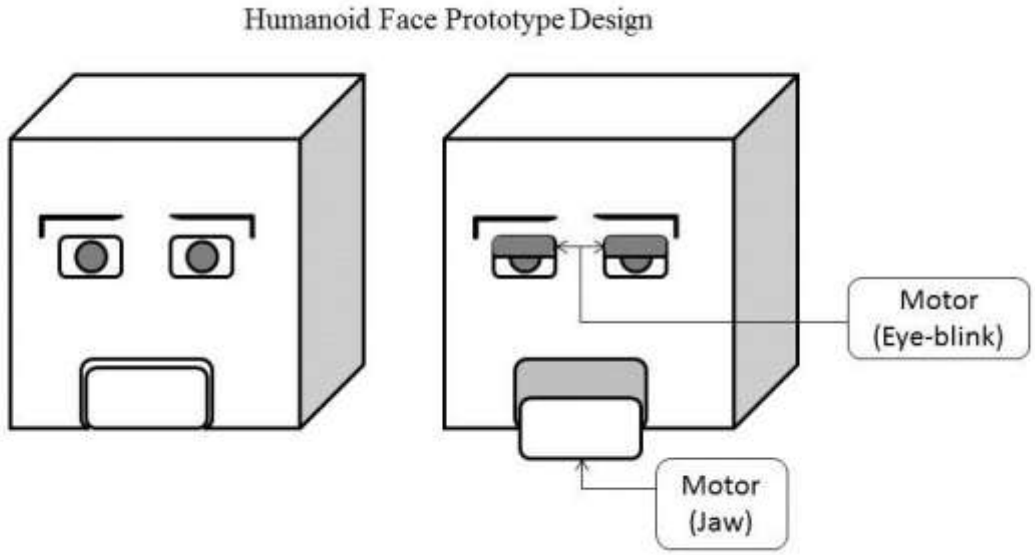
Objectives:

- Real Time facial gesture recognition using OpenCV (two face gestures i.e., Eye Blink and Jaw movement)
- Designing face-prototype using Arduino and Motors to mimic the recognized gestures.

Motivation :

Gesture recognition is an approach for an efficient human-computer interaction which is assumed to be very important in our daily lives. Gestures are meaningful body motions involving physical movements of hands, face etc., In this project, we considered face gestures as our approach to create human-computer interaction. To make the interaction possible, we are designing a humanoid face-prototype to mimic the gestures that are recognized on User’s face. By considering the complexities involved in Hardware & Software, we are limiting to detect only three facial gestures i.e., Eye-Blink and Jaw movement.

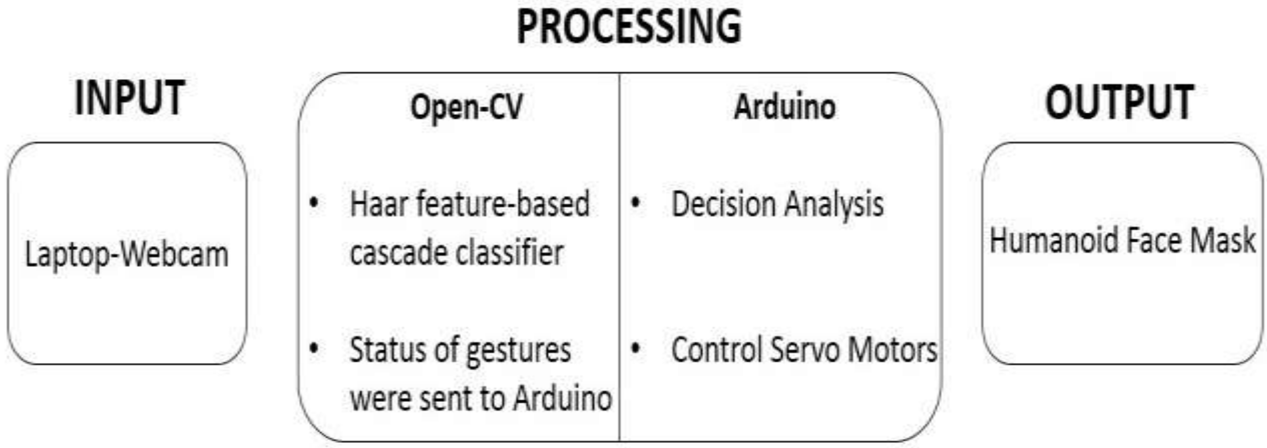
Proposed Hardware Design :



Face prototype (visualising the gestures control by motors)

Description of the Functional Block Diagram and Hardware/Software Components

Functional Block Diagram:



Components:

• Hardware :

1.Arduino
2.Servo Motors (Voltage level: 4.8V – 6V, Current: 0.2A-0.3A)
3. Humanoid face mask

• Software:

1.OpenCV in Ubuntu
2.Arduino Software

Description:

• Input :

Laptop Webcam is used to take the input video.

• Processing :

Processing is done in two modules.

i)Image processing using OpenCV
ii)Arduino processing for motor control

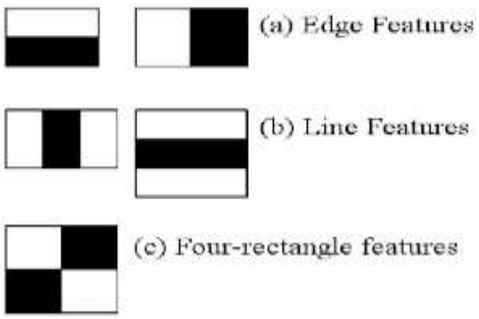
On receiving the video from the webcam, Software code with help of OpenCV library takes input video as frame by frame and analyse the facial gestures in the each frame. Face gesture recognition technique is clearly explained in the following description.

Face Gesture Recognition:

OpenCV uses Viola-Jones method to detect the objects in the image. Viola Jones method has four basic key concepts.

i. Haar features:

Haar features are basically rectangular features. Rectangular features used in Viola Jones method are two-rectangle feature which is the difference between the sum of the pixels within two rectangular regions. These regions are horizontally or vertically adjacent, have the same size and shape. A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle. A four-rectangle feature computes the difference between diagonal pairs of rectangles.



ii. An Internal Image for rapid feature detection:

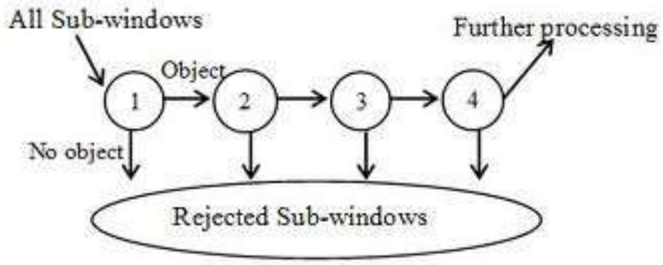
Internal Image technique is used to determine the presence or absence of large number of haar features at every point of the image. Integral value for each pixel is obtained by summing all the pixels above it and to its left. Thus, internal image is created. Using this image, it is easy to find the sum of any rectangular region in the original image. And these rectangular sums would help in finding three kinds of haar features.

iii. Adaboost Machine learning technique:

After finding the features using Internal image technique, we can observe, there are over large number of rectangle features associated with each rectangular region of the image. Number of these features is far larger than the number of pixels. So, it is very difficult to compute the complete set of features for training and classification. An effective classifier can be built using very small number of these features. Finding these features is the major challenge. Viola-Jones method uses AdaBoost Machine learning technique to select set of features and trains the classifier. Generally, AdaBoost algorithm selects a set of weak classifiers to combine and assigns a weight to each of them. This weighted combination is the strong classifier which helps in classifying the objects in the image.

iv. Cascade classifiers:

To increase the detection performance and to reduce computation time, cascade of classifiers are constructed. Each classifier is a separate AdaBoost classifier. A series of classifiers are applied to every sub-window. The initial classifier eliminates a large number of negative examples with very little processing. Subsequent layers eliminate additional negatives but require additional computation. After several stages of processing the number of sub-windows have been reduced radically. Further processing includes other classifiers, if a sub-window passes through all these selected classifiers then it is classified as required object in the image.



Using this procedure, objects in the frame i.e., face, eyes and mouth in the each frame are detected.

After detecting them using OpenCV library, visuals are drawn on the face, eyes and mouth to indicate them. Whenever a face is detected, eyes in that particular face are counted. And this value (either 0 or 1 or 2) is sent to Arduino using USB port serial communication. Arduino observes the sequence of this value continuously. If this value stays ‘0’ for few seconds, then it is indicated as eyes are closed. Then Arduino sends the PWM signal to servo motor to rotate its horn by 90 degrees which closes the eyes of the Humanoid face (shown in the next section). When this value again returns to 2 which means eyes are opened, Arduino sends the signal again to servo motor to rotate back to its normal position. The following pseudo code explains the same for eye gesture mimicking.

Pseudo code to control servo motor attached to eyes:

```
Flag_eyes1 is initialized to 10.
Flag_eyes2 to track the previous state. 11 indicates previous state of eyes were closed and 10 indicates eyes were
opened.

If (eyes closed)    {   Keep track of closed eyes by increment of flag_eyes1 by 1.        }
If (eyes opened)  {   Set the track of closed eyes flag_eyes1 to 10.                        }
If (flag_eyes1 == 13)       // closed eyes for three consecutive frames
                    {
                        Set the flag_eyes2 to 11 as to indicate eyes were in closed state.
                        Set pos_eyes to 90
                        myservo_eyes.write (pos_mouth);       //tells servo to change its position to 90$^0$.

                    }
If (flag_eyes1 ==10 && flag_eyes2 ==11)
                   {
                       Set pos_eyes to 60
                       myservo_eyes.write (pos_mouth);       //tells servo to change its position to 60$^0$.
                       Set the flag_eyes2 to 10 as to indicate eyes were in opened state.
                   }
```

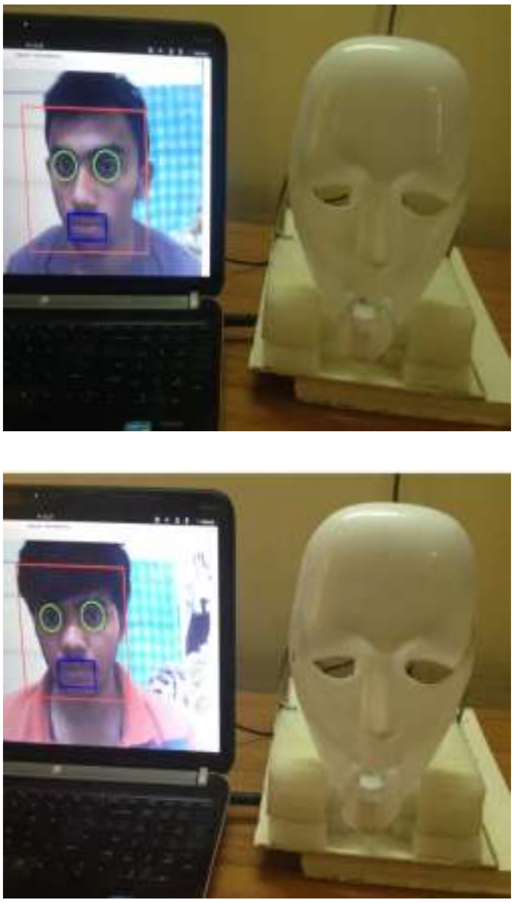
Similarly, in each frame, mouth is detected in the face. Areas of the face cascade object and mouth cascade object are calculated and ratio is obtained. These ratios are obtained for various positions of jaw of multiple users. After the statistical analysis of this data, a certain limit has been kept to discriminate whether mouth is opened or closed. Drawing visuals are adjusted according to these limits. And for every frame, in accordance with this ratio, flag value has been sent to Arduino using USB port serial communication. Arduino observes the sequence of this flag value continuously and sends the PWM signal accordingly to control servo motor attached to mouth of humanoid face prototype. Following pseudo code explains the controlling of motor.

Pseudo code to control servo motor attached to mouth:

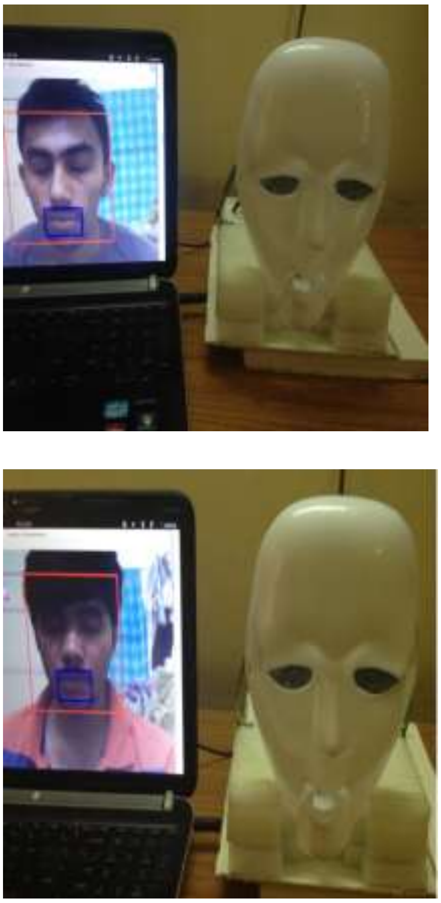
```
Stop_flag to track the previous state. 1 indicates the previous state of mouth was opened and 0 indicates mouth was
closed
If (mouth closed) {   Keep track of closed mouth by increment of flag_mouth1 by 1
                          Set the track of open mouth Flag_mouth2 to 0;
                          if(flag_mouth1>=4 && stop_flag == 1)
                          {                        //Tracking four continuous frames
                              Set pos_mouth to 0
                              myservo_mouth.write (pos_mouth);       //tells servo to change its position to 0$^0$.
                              Set stop_flag to 0;
                          }
                      }
If (mouth opened) {   Keep track of opened mouth for consecutive frames by increment of flag_mouth2 by 1
                          Set the track of open mouth Flag_mouth1 to 0;
                          if(flag_mouth2>=4 && stop_flag == 0)
                          {                        //Tracking four continuous frames
                              Set pos_mouth to 20
                              myservo_mouth.write (pos_mouth);       //tells servo to change its position to 20$^0$.
                              Set stop_flag to 1;
                          }
                      }
```

• Output and Results:

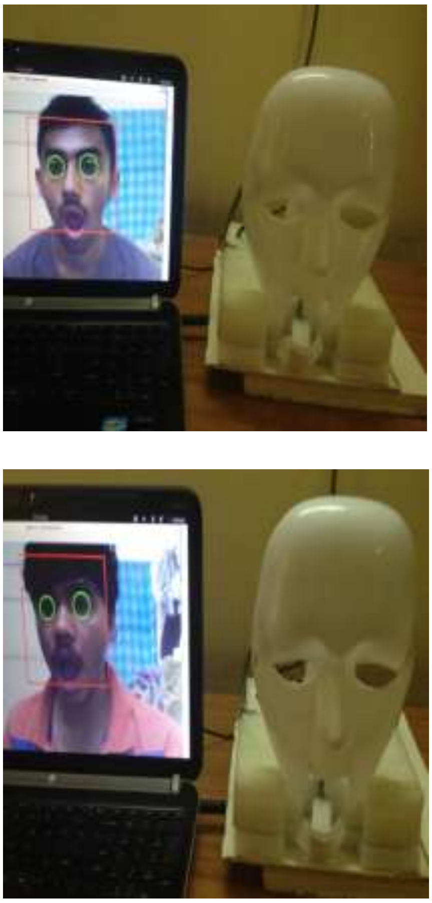
Humanoid face mask is used as base for constructing the whole face prototype. Servo motors controlled by Arduino helps in mimicking the gestures recognized. Eye blink and Jaw movement has been successfully mimicked by Humanoid face prototype.



(i)Images showing
On Laptop: Face is detected, Eyes are detected(open) and mouth detected(closed)
Humanoid face mask: Eyes opened, mouth closed



(ii)Images showing
On Laptop: Face is detected, Eyes are not detected(closed) and mouth detected(closed)
Humanoid face mask: Eyes closed, mouth closed



(iii)Images showing
On Laptop: Face is detected, Eyes are detected(opened) and mouth detected(opened)
Humanoid face mask: Eyes opened, mouth closed

Discussions on the Design with Illustrations

Design of the Humanoid face :

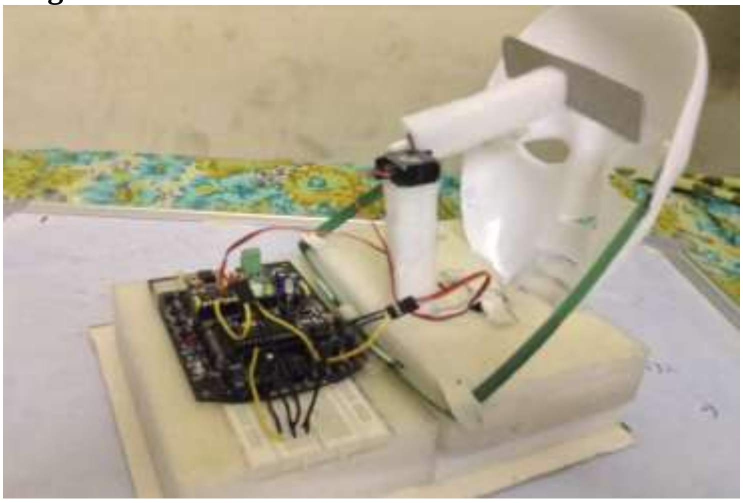


Image showing hardware design of humanoid face (from back side) , servo motors, Arduino

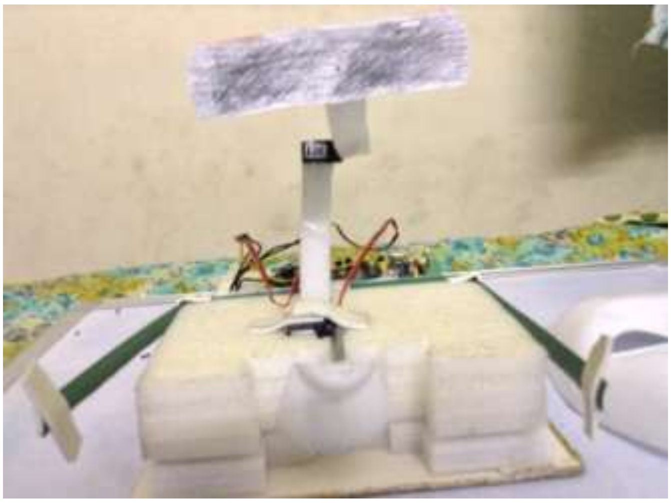
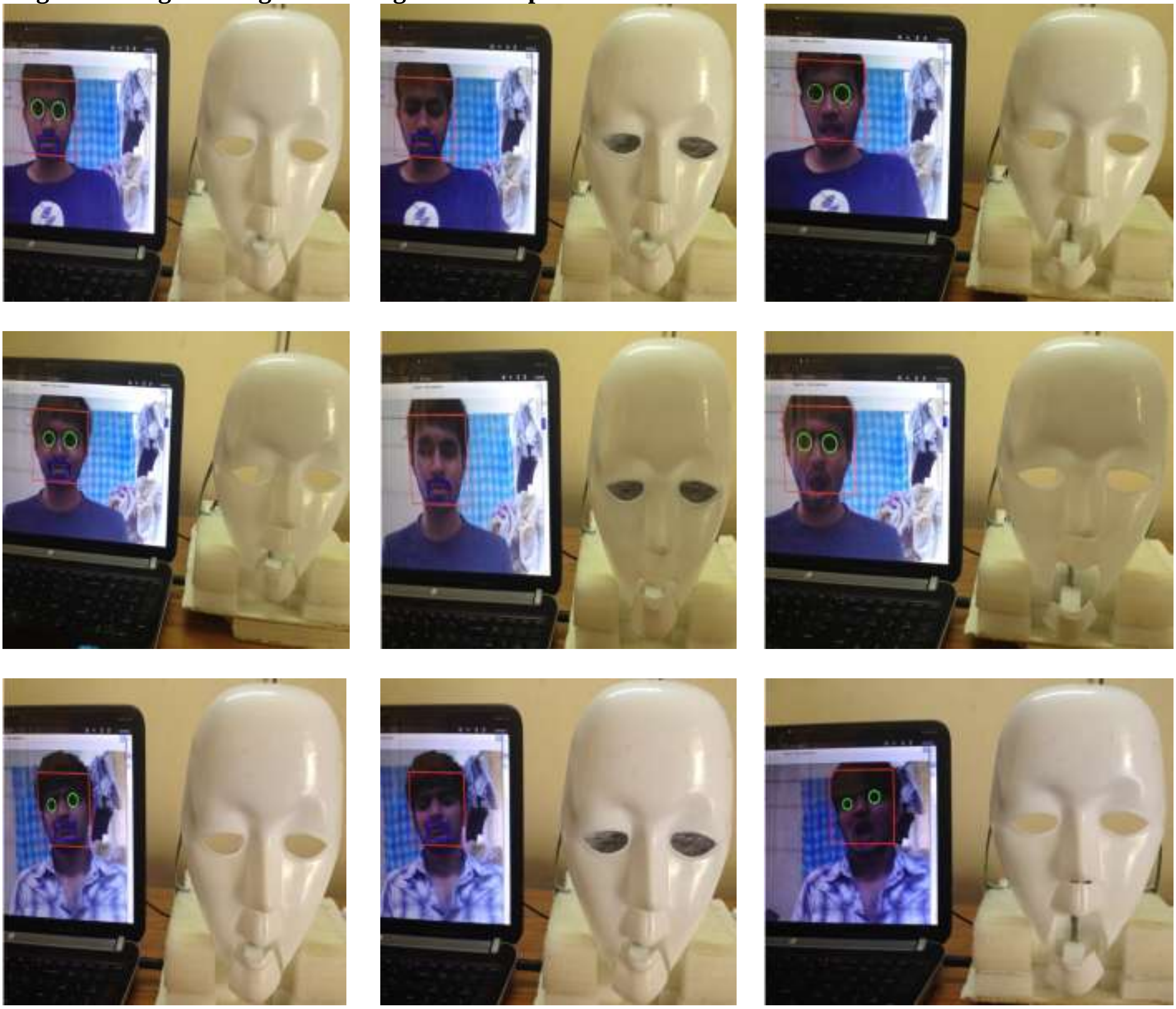


Image showing hardware design of humanoid face (from front side), two servo motors for eyes, mouth

Images showing working of the design for multiple users:



Eyes Opened & Mouth Closed

Eyes Closed & Mouth Closed

Eyes Opened & Mouth Opened

Serial Communication Explanation:

The software code using OpenCV library communicates with the Arduino through a file ‘ttyUSB0’ generated in /dev folder. The software updates the status of flags related to eyes and mouth in this file of Arduino. According to the status of these flag values written in that file of Arduino, Arduino controls the servo motor as explained in the previous section. Hardware design would be much clearer by seeing the following images.

One of the problems faced during serial communication was sending two datasets (values related to eye and mouth) simultaneously. We have overcome this problem by multiplexing the two datasets (shifting one dataset values by 10 units and decoding them accordingly). While the other was providing independent 5V analog signal to two servo motors. Any pin of Arduino cannot be used because those pins cannot provide necessary current for servo motors. Using breadboard has solved this problem.

Summary and References

Summary of the project :

1. Recognition and tracking of the face, eyes and mouth in the streaming video from Laptop Webcam are completed. OpenCV library is used to implement this.
2. Serial Communication between Laptop and Arduino is established. Software output is sent to Arduino for further decision making.
3. Arduino is coded to take decisions such as whether eye & mouth opened or closed.
4. Hardware is designed to mimic the eye blink and jaw movement using mask and servo motors. It is working as expected in good lighting conditions.

The demo video of this project is available in the following link
https://www.youtube.com/watch?v=fJm9DGA1c5s

References:

1. *Paul Viola, Michael Jones*. Rapid Object Detection using a Boosted Cascade of Simple Features in ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001.

List of Components

#	Item Name	Qty.	Provided by (Dept/Self/Guide)	Price (Rs.)
1.	Laptop Webcam	1	Self	
2.	Servo Motors	2	Dept	1000
3.	Arduino	1	Self	
4.	Components to make face prototype		Self	