PROJECT REPORT

Project Name:

Hand Gesture Recognition

Institute Name:
Indian Institute of Technology, Roorkee

Team:

SERIAL NO.	NAME	ENROLLMENT NUMBER
1.	Rohit Kumar	19116056
2.	Rohit Sondarva	19116057
3.	Anubhuti Shruti	19118018

4.	Dhaval Kanani	19112025
5.	Siddhant Taksande	19312032

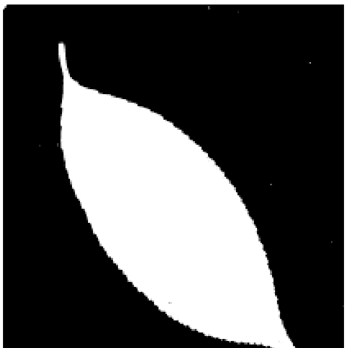
Index

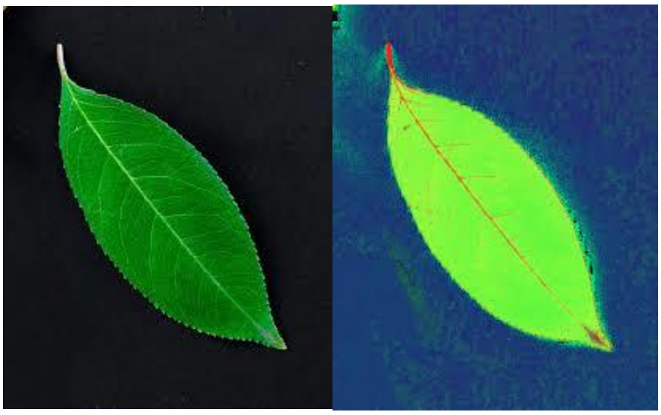
I. Abstract:

This project deals with the detection and recognition of hand gestures. Images of the hand gestures are taken using a web camera and recognised according to certain threshold values. Gesture recognition is one of the essential techniques to build user-friendly interfaces. For example, a robot that can recognize hand gestures can take commands from humans, and for those who are unable to speak or hear, having a robot that can recognize sign language would allow them to communicate with it. Hand gesture recognition could help in video gaming by allowing players to interact with the game using gestures instead of using a controller. However, such an algorithm needs to be more robust to account for the myriad of possible hand positions in three-dimensional space. It also needs to work with video rather than static images. That is beyond the scope of our project.

II. Overview:

Distance Transform: The distance transform is an operator normally only applied to binary images. The result of the transform is a grayscale image that looks similar to the input image, except that the intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point.





Contours: Contours are sequences of points defining a line/curve in an image. Contour matching can be used to classify image objects.

Database: Contains the images of various hand gestures.

Moments: Image moments are useful to describe objects after segmentation. Simple properties of the image, which are found via image moments, include area (or total intensity), its centroid and information about its orientation.

Ratio of the two distance transformed images of the same size = (No of pixels whose difference is zero or less than a certain threshold) / (Total number of pixels in the distance transformed image)

III. Design:

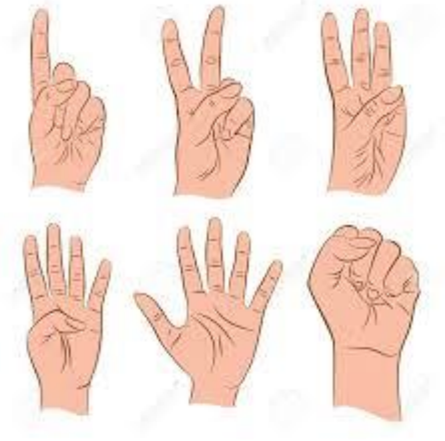
Step 1: User input a picture of the hand to be tested through the Web camera.
Step 2: The image is converted into gray scale and smoothed using a Gaussian kernel.
Step 3: Convert the gray scale image into a binary image. Set a threshold so that the pixels that are above a certain intensity are set to white and those below are set to black.
Step 4: Find contours, then remove noise and smooth the edges to smooth big contours and melt numerous small contours.
Step 5: The largest contour is selected as a target.
Step 6: The angles of inclination of the contours and also the location of the center of the contour with respect to the center of the image are obtained through the bounding box information around the contour.
Step 7: According to certain threshold values of the aspect ratio and counter area, image is recognised and caption is provide according to the result.
Step 8: The results are transferred to the bot through ardunio.

IV. Constraints:

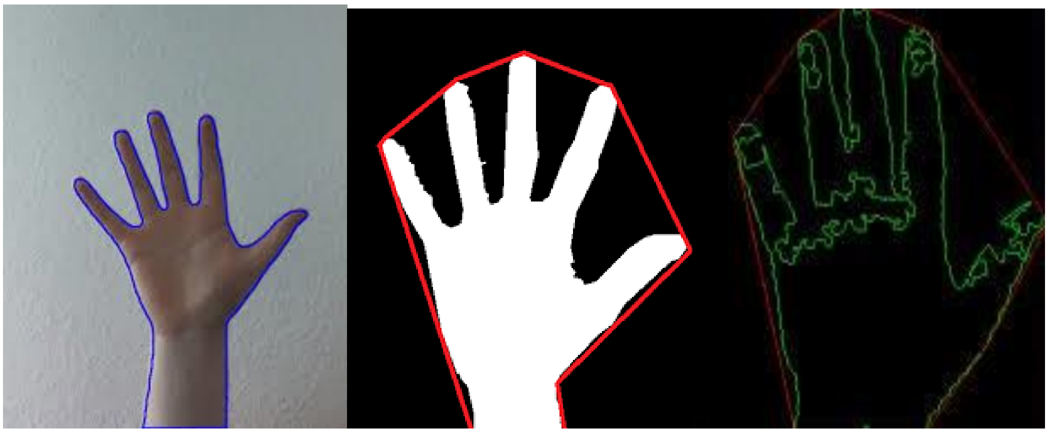
1. The picture of the hand must be taken against a dark background
2. The program recognizes a limited number of gestures.

V. Input Gestures:

In our experiment, we will be identifying limited number of gestures, which is shown as below.



VI. Results:



In the above case, we can see the query image on the left hand side recognise by the program.

VIII. Conclusion:

Based on our observation, we can conclude that the results mainly depend on:
1. Threshold, while converting the gray image to the binary image and finding contours. For example found that uneven lighting across the picture of the hand caused the algorithm to draw contours around the darkened areas in addition to the contour around the hand. Changing the threshold prevented that from happening.
2. The threshold for the Ratio test while matching the distance transformed images. The ratio we used
3. The background, which must preferably be black to get accurate results.

XI. Some Main OpenCV Functions used in the program:

1. To find the number of Contours in the image

int cv2.findContours(. . .)

2. To compute the Convex Hull and Convexity Defects of the Hull

cv2.convexityDefects(. . .)

3. To find bounding box's height, width and center

cv2.boundingRect(. . .)

4. To Find the Counter Area

cv2.contourArea (…)

5. To show images

cv2.imshow (. . .)

XV: Outside Sources

1. The hand gesture images were taken from Google Images.
2. https://youtu.be/uEd2B7fS8Eg
3. https://github.com/vishwajeetsinghrana8/OpenCV/blob/master/Hand_Gesture/Hand_Gesture_Ex2.py
4. https://www.youtube.com/playlist?list=PLkMYhICFMsGajeARsY7N1t1jhbtMb1poL