EECS 332 Final Project

(Prof. Ying Wu)

Finger Cursor

A Hand Gesture Recognition System handling mouse basic operation

Submitted by: -

Aditya Kumar

AKE9173

Contents

Personal views:

I think computer vision is a part of a computer science area which basically deal with images, videos and pixels. It works by analyzing the different components of the image. The core of this course is object identification and object classification in an image or video.

I believe the future needs of this courses are going to increase tremendously. Some of the biggest project and research which are ongoing in this area are like self-driving car, robot visual abilities, image recovery, detection of tumors in the field of health care, etc.

Introduction:

A hand gesture recognition has been a very interesting problem in Computer Vision community for a long time. This is particularly due to the fact that segmentation of foreground object from a background is a challenging problem in real-time. The most obvious reason is because of the semantic gap involved when a human looks at an image and a computer looking at the same image. Humans can easily figure out what's in an image but for a computer, images are just 3-dimensional matrices with their RGB values in it. Therefore, computer vision problems remain a challenge.

Another important challenge is the semantic segmentation problem where a system has to find different regions in an image and tag its corresponding labels in case of an image.

Project description:

This project mainly deals with recognizing the hand gestures from a video sequence or a webcam. To recognize these hand gestures from a live video sequence, we first need to take out the hand region alone removing all the unwanted portions in the video sequence. After segmenting the hand region, we then count the fingers shown in the video sequence and perform some of the basic mouse activity such as left click, right click, moving cursor, and double click based on the finger counts. Thus, the entire problem could be solved using 3 simple steps -

1. Find and segment of the hand region from the video sequence or webcam.
2. Count the number of fingers from the segmented hand region in the video sequence.
3. Perform different mouse activity based on the number of fingers identified.

Design:

The main concepts behind the design of this projects are mentioned as below:

1. Image preprocessing
2. Thresholding
3. Contours detections
4. Convex hulls
5. Convexity defects
6. Finger detection
7. Mouse operations

Image preprocessing:

The first step in hand gesture recognition system is obviously to find the hand region by eliminating all the other unwanted portions by making it black in the video sequence. The video sequence is just a collection of frames or a collection of images that run with respect to time.

After figuring out the background, I bring in the hand and made the system understand that our hand is a new entry into the background which means that it becomes the foreground object but how are we going to take out this foreground alone. The solution is background subtraction. I tried to model the background by telling the system that the first couple of frames is what the background looks like and any changes to this background frame is the foreground.

Finally, I take an image from the video sequence and convert it into the grayscale image. Then I perform a Gaussian blur with a 33x33 convolution matrix on the image for smoothing and to reduce noise from the image. The below thresholding method are performing the background subtraction and differentiating the background from the foreground.

Please note that this project runs best in the complete lighted room with a white background.

Thresholding:

Thresholding is the assignment of pixel intensities to 0's and 1's based on a particular threshold level or value so that our object of interest alone is captured from an image.

To detect the hand region from this difference image, I need to threshold the difference image, so that only our hand region becomes visible and all the other unwanted regions are set as black. This is what Motion Detection is all about.

Another challenge which I faced here is that when the motion has been detected we also have to draw a bounding box and find out whether the foreground object that came out was the hand or was it just a random motion.

In this project the thresholding is done using the Otsu's Binarization method. Otsu's Binarization thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background. It models the background color and the foreground color and creates a histogram of all the colors available in this image. It's a bimodal method so it finds two peeks in the histogram and assign other pixel values based on the nearest peek the value falls marking it as foreground and background.

Contours detections:

Contour is the outline or boundary of an object located in an image. The contour of hand is a series of points which are the boundary pixels of the hand area. In this project, I used OpenCV "cv2.findContours()" in built function with the threshold image as an argument to find the contours. Get the maximum contour area because the contour with the largest area is assumed to be our hand. Please find a sample image with maximum contour are for a hand gesture detection project.

Convex hulls:

Convex hulls are vertices of the smallest convex polygon enclosing the contour. I found the convexity hull for that maximum contour area by using “cv2.convexHull()” function with the maximum contour area value as an argument and draw the convexity hull boundary across the maximum contour area found. Please see below image with maximum contours area detection and convexity hull drawn over it.

Convexity Defects:

Any deviation of the object (contour) from the convexity hull can be considered as convexity defect. In simple words, convexity defect is a cavity in an object (contour) segmented out from an image. That means an area that do not belong to the object but located inside of its outer boundary i.e., convex hull.

The convexity defects for the hand contour are calculated by using the OpenCV built-in function "convexityDefects()" passing the contour and convex hull values as an argument.

Finger detections:

After the convexity defects are obtained, I perform steps for identifying the fingertips and the fingers. Convexity defects obtained, is a structure that returns four values, start point, end point, farthest point and approximate distance to farthest point. The formula below, denotes for one of the contours the start, the end and the far point. C represents the start point, B represents the end point and A is the farthest point. The angle made by the two fingers must be found to correctly determine if a finger is held up. This is done using the triangle formed by A, B and C. The length of each line is found using the distance formula as

$$a = \sqrt{(start[0] - end[0])^2 + (start[1] - end[1])^2}$$
$$b = \sqrt{(start[0] - far[0])^2 + (start[1] - far[1])^2}$$
$$c = \sqrt{(end[0] - far[0])^2 + (end[1] - far[1])^2}$$

Once the length of each have been found, using the Cosine rule,

$$a^2 = b^2 + c^2 - 2bc\cos A$$

the angle A is found using

$$A = \cos^{-1}(b^2 + c^2 - a^2/2 * b * c)$$

If the angle A is less than 90°, it is considered that two fingers are held up. The same technique is applied on all the defects to find the number of fingers held up.

The approach can only be used to find the number of fingers and if the number of fingers held up are more than two.

Mouse operations:

I used the inbuilt python mouse library for performing basic mouse operations, like cursor movement, left click, right click, double click etc. The main concept used here is number of convexity defects found in the give sub windows created for hand detection.

The system performs below mouse operation based on number of convexity defects founds.

1. If there are no convexity defects, then the system performs mouse cursor movement then it is assumed that only single finger is raised, and the system perform cursor movement on the same.

2. If there is one convexity defects, that means the two of the fingers is raised and the system performs mouse left click operation.
3. If there are two convexity defects, that means the three of the fingers is raised and the system performs mouse double click operation.
4. If there are three convexity defects, that means the four of the fingers is raised and the system performs mouse right click operation.
5. If there are more than four convexity defects, that means the entire hand is raised and the system does not perform any operations.

Implementation:

Algorithm (Code Flow):

The following algorithm shows the basic steps performed by the system when gestures are detected.

1. Start the webcam.
2. Created a sub window for capturing the hand, it will not detect hand from entire screen. Hands need to be present in this sub window.
3. Capture the image
4. Convert the image to gray scale
5. Perform Gaussian Blur on the image for smoothing and to reduce noise from the image.
6. Perform Thresholding using the Otsu's Binarization method.

7. Extract the contours using “cv2.findCountours()” in built function depending on the cv2 versions.
8. Find the maximum contour area using “cv2.contourArea()” for identifying the hand gesture as we know that hand gesture will have the maximum area.
9. Find the topmost point in the maximum contour area found in the last step for finding the index finger.
10. Create bounding rectangle around the max contour area.
11. Find the convex hull around the max contour area and draw the same.
12. Find the convexity defects using cv2.convexityDefects for applying the cosine rule on that to identify the number of fingers in the contours we found in the step 8.
13. Run below code inside a for loop for each defect found in the previous steps
    a. Apply the Cosine Rule to find angle for the defects (between fingers).
    b. If the angle is less than or equal to 90 then increase the number of defects by one.
14. If the number of convexity defects is 0, then I assume that the number of fingers is only one and perform cursor movement in this case.
15. Else If the number of convexity defects is 1 then then there will be 2 fingers identified and perform the left click operation in this case.
16. Else If the number of convexity defects is 2 then there will be 3 fingers identified and perform the right click operation in this case.
17. Else If the number of convexity defects is 3 then there will be 4 fingers identified and perform the operation of double click in this case.

18. Else if the number of convexity defects is more than 4 then we assume that the complete hand is detected and no operation is performed in this case.
19. Keep performing the above steps unless “ESC” button is pressed.
20. If “ESC” button is pressed, then close all windows and close the camera.

Result and Analysis:

Prerequisite:

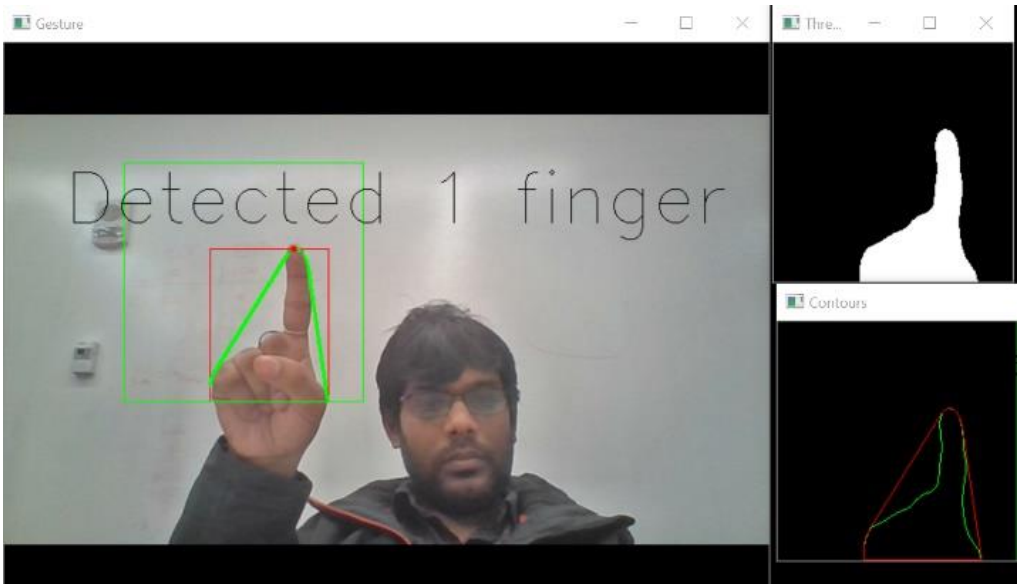
- Below results are generated in complete lighted room with a white background.
- Hand should be present in predefined sub domain square to perform different mouse operations.

Some of the result of my applications are shown below. There are three images in each result image consist of following images:

1. Main hand gesture image with number of fingers detected written on it.
2. Threshold image
3. Contours image with convex hull drawn with red boundary.

Result 1: Finding one index finger:

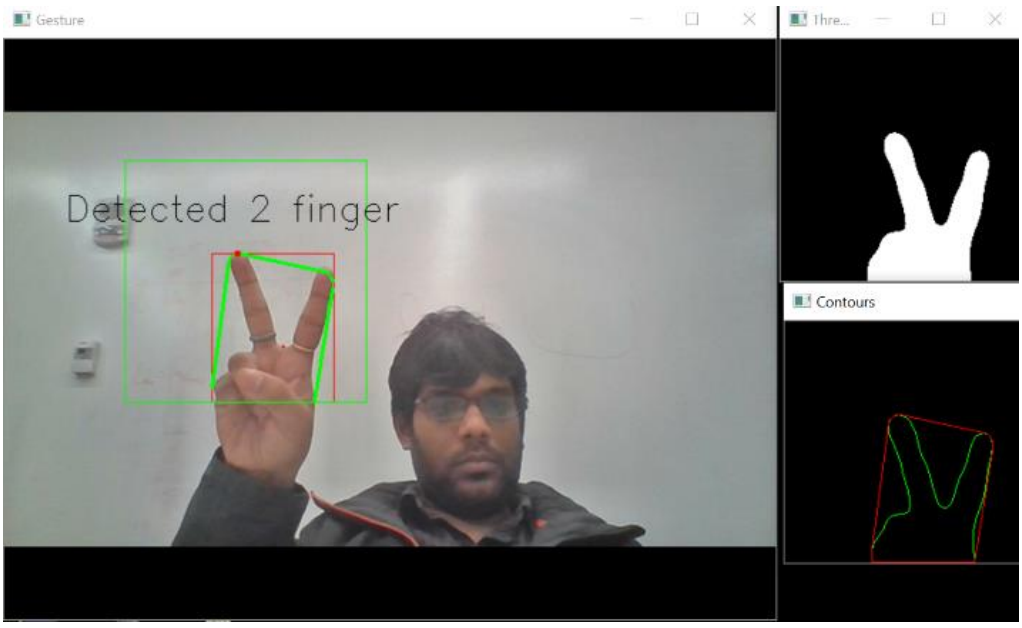
The below image clearly shows the topmost point (dark red circle) of the index finger. The movement of this point inside the green square can make your mouse cursor movement.



Result 2: Finding two fingers and one convexity defect:

The below image clearly shows the one convexity defect (light red circle) between two fingers.

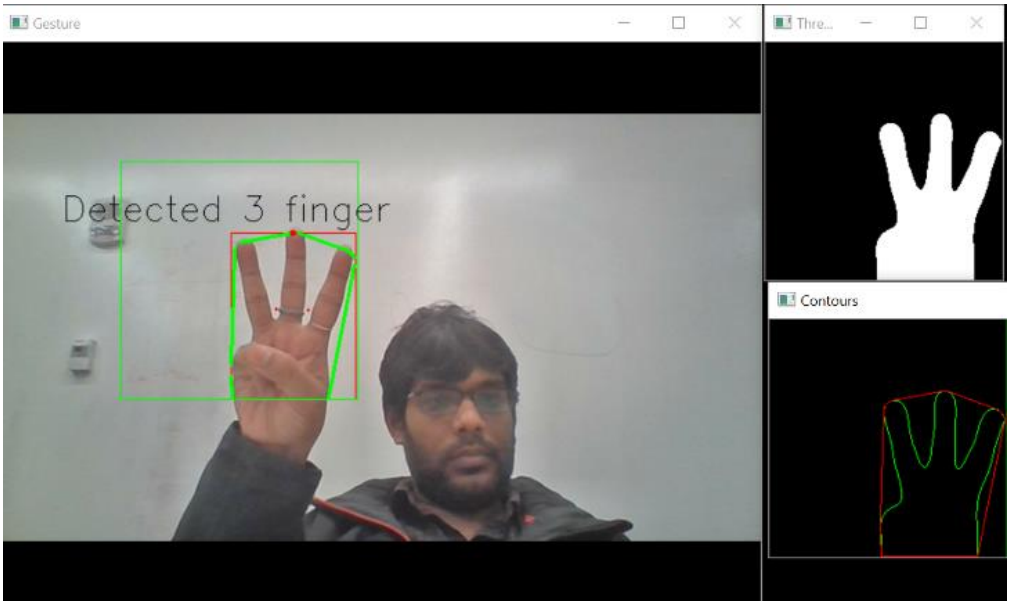
This can perform a left click of the mouse operation on the screen.



Result 3: Finding three fingers and two convexity defects:

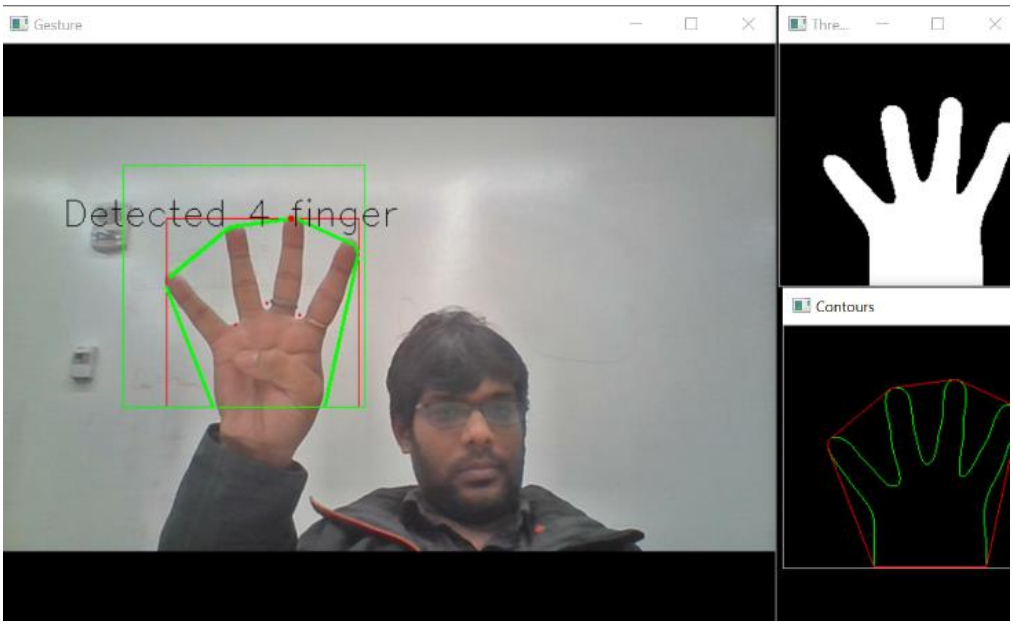
The below image clearly shows the two convexity defects (light red circles) between three fingers.

This can perform a right click of the mouse operation on the screen.



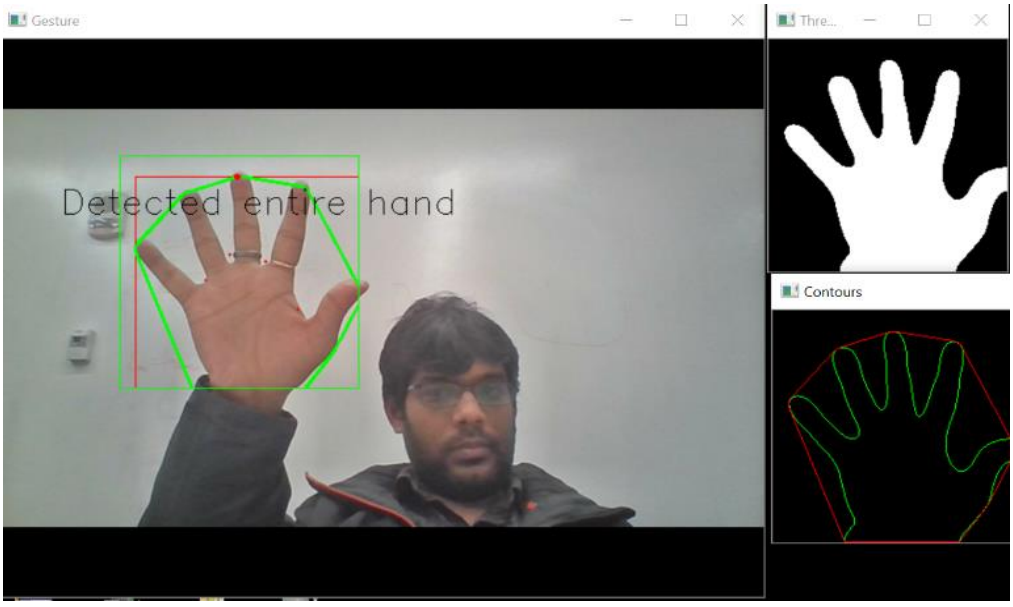
Result 4: Finding four fingers and three convexity defects:

The below image clearly shows the three convexity defects (light red circles) between four fingers. This can perform a double click of the mouse operation on the screen.



Result 5: Finding entire hand with four convexity defects:

The below image clearly shows the four convexity defects (light red circles) between five fingers.



Remarks:

This project describes a system that controls computer applications with the help of hand gestures. The method proposed here successfully created a hand gesture recognition system, that can recognize which gesture is performed by the user and accurately perform the functionality associated with it.

Presently, the webcam, microphone and mouse are an integral part of the computer system. This system uses only webcam which can eliminate the mouse. Also, this would lead to a new era of Human Computer Interaction where no physical contact with the device is required.

Future Works:

The current system gives best results in a plain white background and hence puts certain constraints on the user for successful working. The future works can be done on this project to handle any background in any lighting condition for making it more robust.

Currently, the system detects the objects (Hands and fingers) in a sub window only. Future works can be done on this project to cover complete webcam window for detecting the hands region and fingers.

One of the major flaws in the current project is detecting contours and convexity defects in case of any other objects are identified in the given sub window. It detects some random number of convexity defects and perform some mouse operation accordingly. I tried to limit the maximum contour area for fixing the same but did not able to achieve must success. So, Future works can be done to fix this major issue.

Another future work will include implementation of additional gestures which will enable the user to perform more operation with ease. The proposed system uses only the one hand to perform gestures. Hence, enhancement of the technique proposed, is possible using both hands for performing different computer operations.

Experiments and extensive testing need to be done on a larger scale so that results can be more accurate, and system can become more scalable.

Course feedback and suggestions:

This course has given me the preliminary understanding on the different fields and the basic concepts of Computer Vision. Moreover, I learned about the mathematical principles behind the important concepts in image and video processing. Before taking course, I was new to this field and after completing this course, I feel that I have learned a lot about the basics concepts of computer vision which I can apply in the industry and/or developing something fun project of my own in my free time.

Professor Ying Wu is passionate about the topics covered in class and he always used some interesting examples and stories to attract us and illustrate the basic concepts. Moreover, he was very kind and patient with the queries and confusion which I had in the lecture. He gave valuable suggestions to me and inspired me on the ideas of some cool projects.

On the aspect of the course organization and assignments, it wonderfully designed and all assignments are perfectly timed which help us to refresh the memory as well as better understand the knowledge we learned from the lectures.

Two suggestion which I would like to provide are mentioned below: -

1. I think it would be great if the course can have a Piazza website where we can post our questions and get answers. I have tried in many of my subjects and it was very successful.
2. I am not sure if this suggestion is going to be implemented but it will be good if the classes are recoded and made available to students and alums of the university. So that everyone can brush up their concepts whenever required.

Personally, I really like this course and I will recommend it to my friends who has interests in Computer Vision. The course itself include a wide range of topics in depth and it also provides students the capability to implement their ideas using techniques they learned from the course.