IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

A TECHNIQUE FOR HAND GESTURE RECOGNITION

Amrita Bhatnagar

AP, CSE Dept, Akgec
Ghaziabad, U.P India

Abstract

Hand gesture recognition is one of the systems that can detect the gesture of a hand in a real-time video. Hand Gesture Recognition could be used in various areas like in shopping complexes, controlling different operations in a computer like a mouse or media player, etc. It gives the desired output according to the input given with our fingers. The gesture of hand is classified within a certain area of interest. In this study, designing hand gesture recognition is one of the complicated jobs that involves two major problems. Firstly, is the detection of hand. Another problem is to create a sign that is suitable to be used one hand at a time. This paper concentrates on how a system could detect, recognize and interpret hand gesture recognition through computer vision with the challenging factors which are variability in pose, orientation, location, and scale. To perform well in developing this paper, different types of gestures such as numbers and sign language need to be created in this system. In this paper, the detection of hand will be done using the theories of Region of Interest (ROI) via Python programming.

Keywords: Gesture, Region of Interest (ROI), Frame, Thresholding, Contour, Convexity

1. Introduction

In day-to-day life, hand gesture recognition is one of the systems that can detect the gesture of a hand in a real-time video. The gesture of hand is classified within a certain area of interest. Designing a system for hand gesture recognition is one of the goals of achieving the objectives of this paper. The task of recognizing hand gestures is one of the main and important issues in computer vision. With the latest information and media technology advances, human-computer interaction (HCI) systems involve hand-processing tasks such as hand detection and hand gesture recognition.

The first step in any hand processing system is to detect and locate the hand in the real-time video from the webcam. The detection of hands is challenging because of pose, orientation, location, and scale variations. Also, different intensity of light in the room adds to the variability. In the process of detection of hand, Hand gesture recognition generally involves multiple levels such as image acquisition, pre-processing, feature extraction, and gesture recognition. Image acquisition involves capturing image in the video frame by frame using a webcam. The captured images go through the image pre-processing process which involves color filtering, smoothing, and thresholding.

In this study, designing hand gesture recognition is one of the complicated jobs that involves two major problems. Firstly, is the detection of hand. The user's hand is detected by using a webcam in real-time video. The problem would be the unstable brightness, noise, poor resolution, and contrast.

The detected hand in the video is recognized to identify the gestures. At this stage, the process involves segmentation and edge detection. Various information of images like color, hand posture, and shape-based (shape of a hand) in real-time would affect the recognition of gestures. Another problem is to create a sign that is suitable to be used for one hand at a time. The extraction of hand needs to be followed to determine each number and sign used. The extraction of the hand involves the contour and convexity defects.

Convexity defect gives an issue on how to calculate the depth of the defects. Some defects have far greater depth than others, so identifying the depth would include some equations.

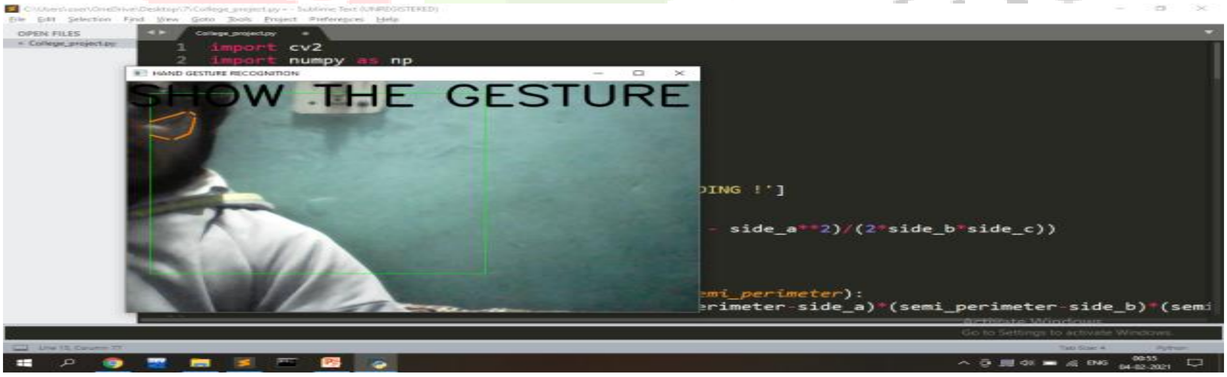
The list of objectives that will need to be achieved for this paper: to establish a complete system for detecting, recognizing, and interpreting hand gesture recognition through computer vision using Python and OpenCV, and to create the numbers and sign languages of hand gestures shown in the system that will meet the name of the paper. This study comprises how to implement a complete system that can detect, recognize and interpret the hand by using Python and OpenCV in any intensity of light, pose, or orientation of the hand. In order to accomplish this, a real-time gesture-based system is developed. In the proposal stage, designing a system that could detect the hand through the contour of the hand. The contour of the hand refers to the curve for two variables for the function along which that function has a contact value that is not changed. Besides, to detect the appearance of a hand in a frame of the real-time video. For the hardware implementation section, a wired web camera is used thus hand gesture recognition can be done and implemented. The direction of the web camera is important to obtain a good video without any shadowing effects and the intensity of light in a room is enough. By selecting the correct light intensity and direction, the shadows can be reduced to a minimum.

2. Methodology

In this paper Hand gesture recognition is done using Python programming language and Open CV as library. Python programming language produces simple and easy system code to understand. The image that is captured using a web camera will be processed in a region called as Region of Interest (ROI) which acts as a region of the wanted area while ignoring the outside region, called background and background also should not have any part of hand.

In general, the selection of an appropriate segmentation algorithm depends largely on the type of images and the application areas. This only work when a frame captures a frame from the video and converts the RGB format into HSV. Region of Interest (ROI) Region of Interest, or commonly abbreviated as ROI, is a region that will be considered as the region of wanted area while ignoring the outside region, called background. To detect the availability of hand recognition, the detected region must be located the same as the region of interest. In other words, as the classifier tracks the wanted object, in this case, the bounding rectangle around the hand contour, the area of the hand must overlap with the region of interest.

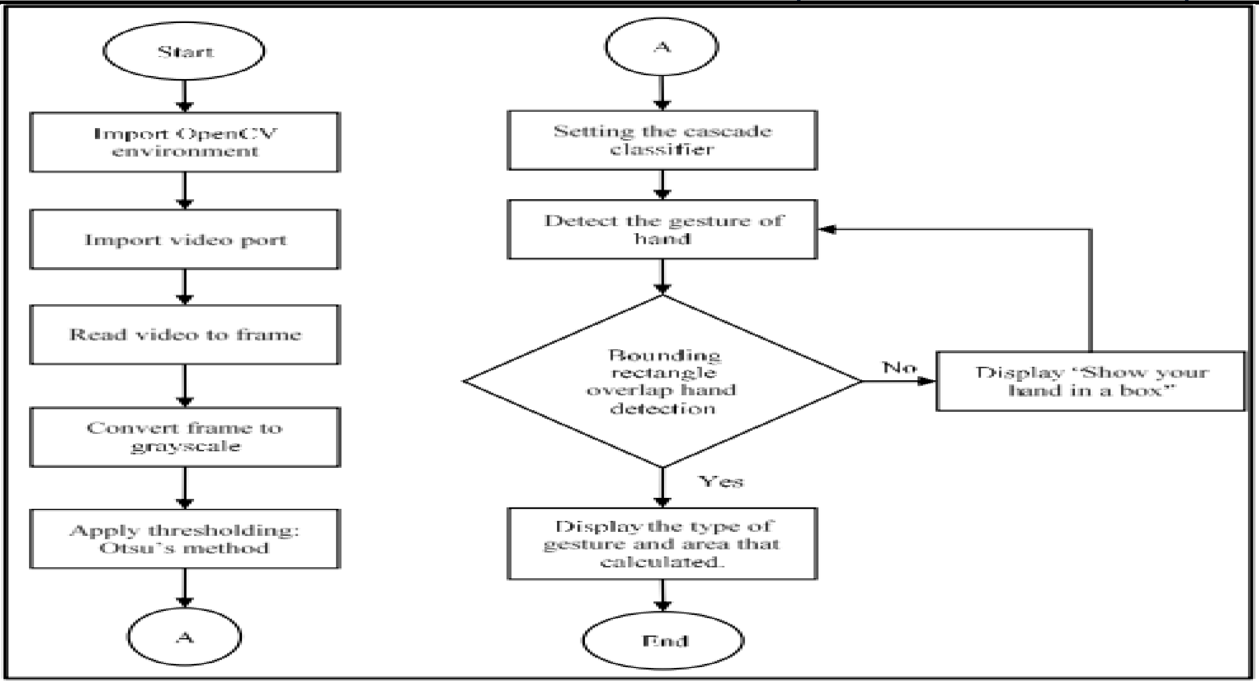
By using the conditional method of overlapping region, the system can identify whether the region selected, in this case, the rectangle around the hand contour, overlaps with another active region, in this case, the hand itself. By developing this theoretical part of ROI to the simulation and hardware implementation, the process of detecting the appearance of the hand will be much simplified compared to the other method.



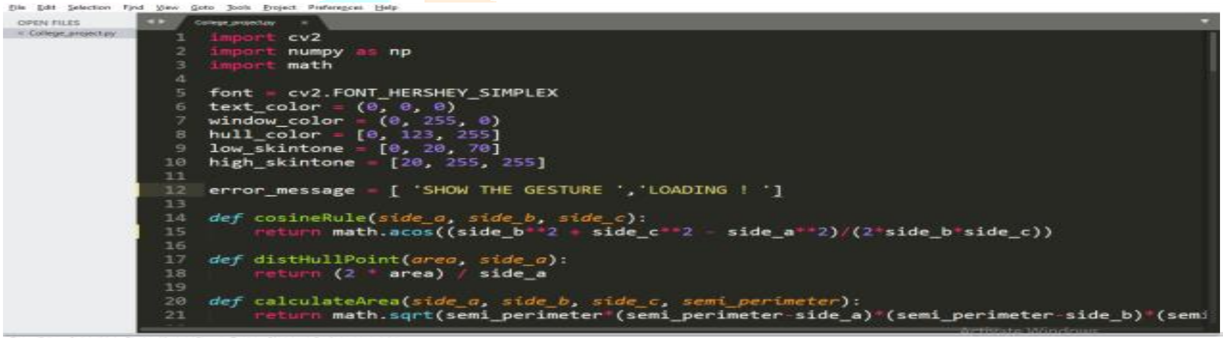
```
import cv2
import numpy as np
DING !']
- side_a**2)/(2*side_b*side_c))
emi_perimeter):
erimeter-side_a)*(semi_perimeter-side_b)*(semi
```

The figure shows the criteria for ROI to overlap that can be considered as overlapping ROIs. These criteria are needed to make sure that the program can identify the detection of hand gestures.

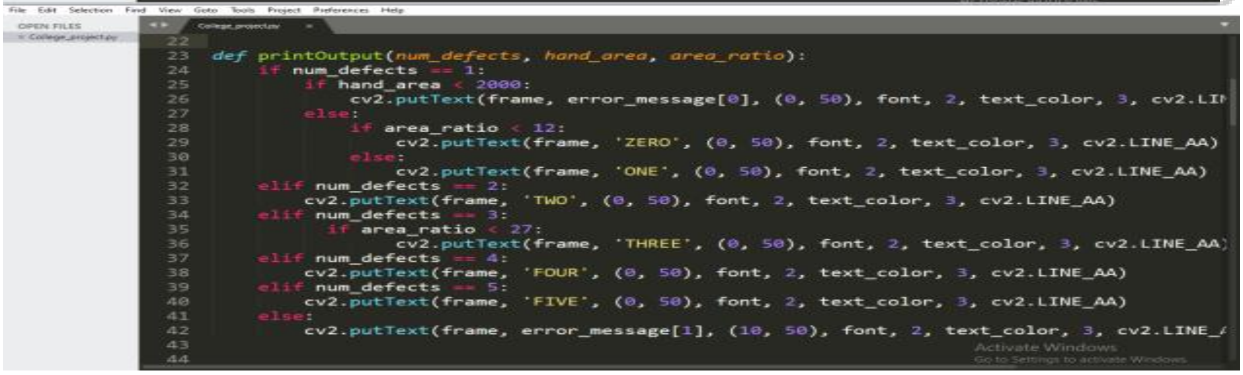
For this paper, the Region of Interest (ROI) is used to see the gesture of hand because the implementation for the ROI is suitable to detect the overlapping two regions on it. Besides, the system will detect only two outputs for each hand gesture recognition Application of Region of Interest (ROI). Region of Interest (ROI) Region implemented for ROI. An active region, in this case, the hand itself. ROI2 The rectangle sub-window on the screen. This region is static or not movable outside the frame of the real-time video and it highlights the area that involves.


Start
Import OpenCV environment
Import video port
Read video to frame
Convert frame to grayscale
Apply thresholding: Otsu's method
A
A
Setting the cascade classifier
Detect the gesture of hand
Bounding rectangle overlap hand detection
No
Display "Show your hand in a box"
Yes
Display the type of gesture and area that calculated.
End

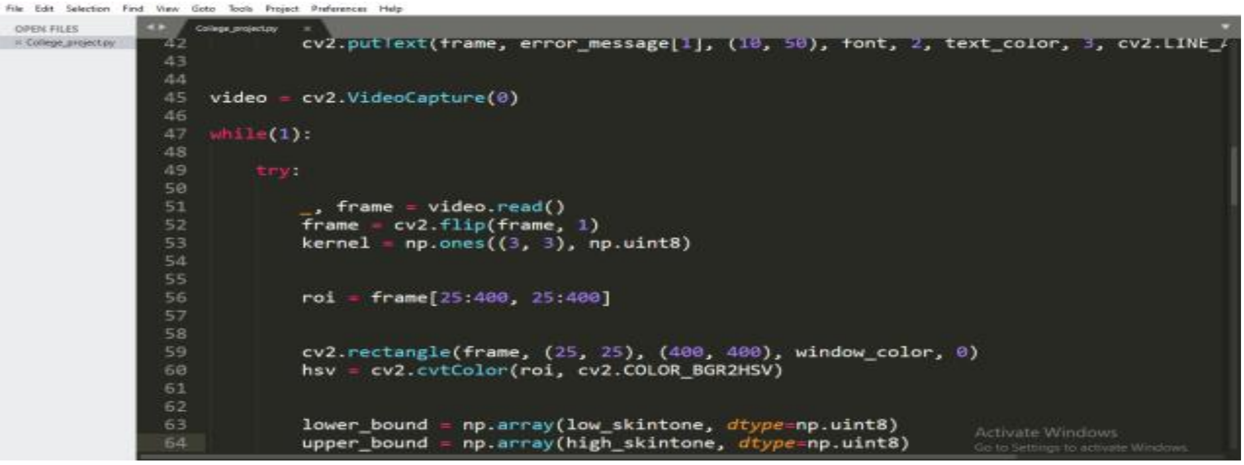
3.Implementation



```python
import cv2
import numpy as np
import math

font = cv2.FONT_HERSHEY_SIMPLEX
text_color = (0, 0, 0)
window_color = (0, 255, 0)
hull_color = [0, 123, 255]
low_skintone = [0, 20, 70]
high_skintone = [20, 255, 255]

error_message = [ 'SHOW THE GESTURE ','LOADING ! ']

def cosineRule(side_a, side_b, side_c):
    return math.acos((side_b**2 + side_c**2 - side_a**2)/(2*side_b*side_c))

def distHullPoint(area, side_a):
    return (2 * area) / side_a

def calculateArea(side_a, side_b, side_c, semi_perimeter):
    return math.sqrt(semi_perimeter*(semi_perimeter-side_a)*(semi_perimeter-side_b)*(semi
```



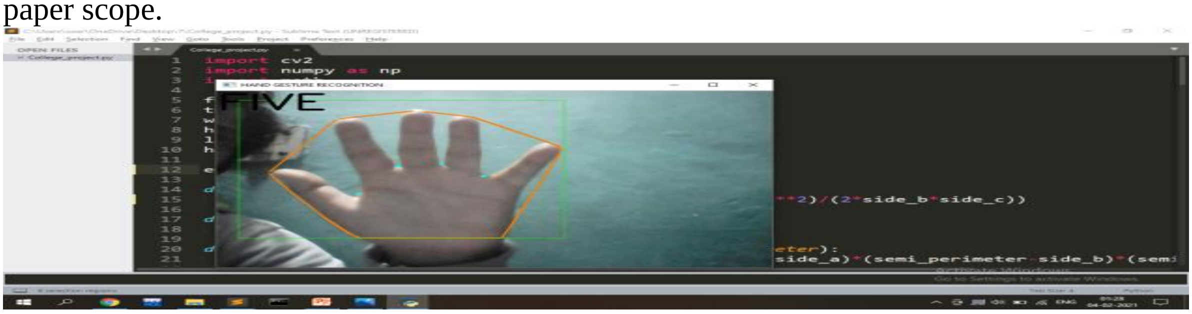
```python

def printOutput(num_defects, hand_area, area_ratio):
    if num_defects == 1:
        if hand_area < 2000:
            cv2.putText(frame, error_message[0], (0, 50), font, 2, text_color, 3, cv2.LIN
        else:
            if area_ratio < 12:
                cv2.putText(frame, 'ZERO', (0, 50), font, 2, text_color, 3, cv2.LINE_AA)
            else:
                cv2.putText(frame, 'ONE', (0, 50), font, 2, text_color, 3, cv2.LINE_AA)
    elif num_defects == 2:
        cv2.putText(frame, 'TWO', (0, 50), font, 2, text_color, 3, cv2.LINE_AA)
    elif num_defects == 3:
          if area_ratio < 27:
                cv2.putText(frame, 'THREE', (0, 50), font, 2, text_color, 3, cv2.LINE_AA)
    elif num_defects == 4:
        cv2.putText(frame, 'FOUR', (0, 50), font, 2, text_color, 3, cv2.LINE_AA)
    elif num_defects == 5:
        cv2.putText(frame, 'FIVE', (0, 50), font, 2, text_color, 3, cv2.LINE_AA)
    else:
        cv2.putText(frame, error_message[1], (10, 50), font, 2, text_color, 3, cv2.LINE_


```



```python
        cv2.putText(frame, error_message[1], (10, 50), font, 2, text_color, 3, cv2.LINE_


video = cv2.VideoCapture(0)

while(1):

    try:

        _, frame = video.read()
        frame = cv2.flip(frame, 1)
        kernel = np.ones((3, 3), np.uint8)

        roi = frame[25:400, 25:400]


        cv2.rectangle(frame, (25, 25), (400, 400), window_color, 0)
        hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)


        lower_bound = np.array(low_skintone, dtype=np.uint8)
        upper_bound = np.array(high_skintone, dtype=np.uint8)
```

4. Result and Discussions

The explanation of the results will be focused on the simulation part since the difference in the hardware implementation is the source code of the real-time input video. These results are analyzed throughout the paper scope.



```
import cv2
import numpy as np
**2)/(2*side_b*side_c))
eter):
side_a)*(semi_perimeter-side_b)*(semi
```

The figure shows how a convex hull is applied to find the edge of the user's hand. Initially, the convex hull compares with hand contours to detect convexity defects where defects are the point. where both are different. Firstly, the vertices of the convex hull are calculated where they change direction. Then, the hand contour segment between consecutive pairs of consecutive vertices of convex hull and finds that pixels in each segment maximize the distance from the convex hull. This maximum distance is called the defect depth where the dots are called convexity defects. The proposed hand gesture recognition requires a webcam to capture the user's gesture, preferably with clear background. The test is done under normal lighting, the system can recognize and track the user's hand movement. The segmentation process is able to detect the user's movement and count the contours. Each signal differentiates using different ROI. The image above shows the output when execution of Python file However, there are some limitations of the program, i.e. the lighting environment changes are still influential in the segmentation process, in particular the process of background removal. Also, the user's hand and the webcam must be in a fixed position so the webcam will capture the image with the exact area to identify the gesture.

5. Conclusion

We were able to create a robust gesture recognition system that did not utilize any markers, hence making it more user-friendly and low-cost. In this gesture recognition system, we have aimed to provide gestures, covering almost all aspects of HCI such as system functionalities, launching of applications, and opening some popular websites. For future recommendations, this system will include the execution of additional gestures that will allow any users with different skin colors and sizes of palm to perform more functions easily. The current system only uses the right hand with a specific area in ROI to perform gestures. Additionally, background subtraction algorithms can be used for more effective performance.

References

[1] Granit Luzhnica, Elizabeth Lex, Viktoria Pammer. A Sliding Window Approach to Natural Hand Gesture Recognition using a Custom Data Glove. In: 3D User Interfaces (3DUI); 2016 IEEE Symposium on ; 2016 Mar 19 ; New York : IEEE; 2016 ; p.81-90.

[2] Mitra, S.; Acharya, T. Gesture recognition: A survey. IEEE Trans. System Man Cybern. Part C Appl. Rev. 2007, 37, 311–324.

[3] N. Otsu 1979 A Threshold Selection Method from Gray-Level Histograms (IEEE Trans. Syst.Man. Cybern.) vol. 9 no. 1 pp 62–66.

[4] Ji-Hwan Kim, NguyenDuc Thang, Tae-Seong Kim 3-D hand Motion Tracking and Gesture Recognition Using a Data Glove Industrial Electronics; 2009 IEEE International Symposium on ; 2009 July 5, IEEE, New York (2009), pp. 1013-1018

[5] Hung CH, Bai YW, Wu HY. Home appliance control by a hand gesture recognition belt in LED array lamp case. In: Consumer Electronics (GCCE); 2015 IEEE 4th Global Conference on ; 2015 Oct 27; New York : IEEE;2015; p. 599-600.

[6] Dulayatrakul J, Prasertsakul P, Kondo T, Nilkhamhang I. Robust implementation of hand gesture recognition for remote human-machine interaction. In :Information Technology and Electrical Engineering (ICITEE); 2015 7th International Conference on ;2015 Oct 29; p. 247-252.

[7] Monochrome glove: A robust real-time hand gesture recognition method by using a fabric glove with design of structured markers
Virtual Reality (VR); 2016 IEEE ;2016 Mar 19; Greenville, SC, IEEE, New York (2016), pp. 187-188

[8] https://iopscience.iop.org/article/10.1088/1757-899X/99/1/0120

[9] G. R. S. Murthy, R. S. Jadon. (2009). "A Review of Vision Based Hand Gestures Recognition," International Journal of Information Technology and Knowledge Management, vol. 2(2), pp. 405- 410.

[10] Xingyan Li. (2003). "Gesture Recognition Based on Fuzzy C-Means Clustering Algorithm", Department of Computer Science. The University of Tennessee Knoxville.

[11] Mahmoud E., Ayoub A., J¨org A., and Bernd M., (2008). "Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition", World Academy of Science, Engineering and Technology 41.

[12] 10. M. M. Hasan, P. K. Mishra, (2011). "HSV Brightness Factor Matching for Gesture Recognition System", International Journal of Image Processing (IJIP), Vol. 4(5).

[13] N. A. Ibraheem., R. Z. Khan, (2012). "Vision Based Gesture Recognition Using Neural Networks Approaches: A Review", International Journal of Human Computer Interaction (IJHCI), Malaysia, Vol. 3(1).

[14] P. Garg, N. Aggarwal and S. Sofat. (2009). "Vision Based Hand Gesture Recognition,"

[15] Aashni Haria, Archanasri Subramanian, Nivedhitha Asokkumar, Shristi Poddar, Jyothi S Nayak,Hand Gesture Recognition for Human-Computer Interaction, Procedia Computer Science, Volume 115,2017, Pages 367-374, ISSN 1877-0509