Convex hull based gesture recognition calculator using OpenCV

Pavankumar.S
School of Computer Science and Engineering (SCOPE),
VIT University,Vellore,India
pavan_engg@icloud.com

Hariharan.R
School of Computer Science and Engineering (SCOPE),
VIT University,Vellore,India
hariharan.rm3@gmail.com

Abstract:

Gestures have been existent since the pre medieval times and have served as an important means of communication. With the growth of computing power and technologies, computers can now perform complex mathematical calculations at faster pace and improved efficiency. In spite of the advancement in the computing industry, the input has to be still given traditionally through the keyboard and the gesture based input system still remains tweak. The objective of this paper is to use hand gestures as an input to solve some mathematical expressions, the gestures can be further extended for other purposes like minimising the window and navigating between the screen based on the need. The gestures are then suitably transformed into different numbers using the convex hull mapping for different gestures. The convex hull is the minimal polygon that encloses a given set of points. The proposed model works without help of any Machine learning algorithms, as our model is expected to be computationally quicker and is also to be made lightweight so that we could extend its usage to devices with minimal computational power.

I. INTRODUCTION

Human Computer Interaction(HCI) and Computer vision industries are closely correlated[4]. The ways in which we interact with computer have been widened with the help of the computer vision industry. E- commerce industries have got a newer source of data in the form of gestures, apart from the pointer speed and hovering patterns that were earlier tracked to provide recommendations for the user. By using gestures as an input one can reduce the dependence on other peripheral devices of the computer. Gestures are also easier to learn and can be used by anyone, even those without the knowledge on operation of computers.

The primary goal of proposed system is to identify few hand gestures and convert to a form such that a computer can actually recognise the given input and respond accordingly. An alternate approach with a colored glove on hand has been worked upon to detect gestures but it had to be disregarded, since it required highly controlled environment for its working. It used HSV based color segmentation for its working, so we had to rely on a coloured object(a glove, ball , pen etc..) to give input. Other limitations in the model include the misinterpretation of similarly colored object in the background as controller, variation in the light intensities due to daylight etc..

Fig.1 Shows the original image(left hand side) for extracting mask(ROI) using HSV segmentation and the segmented image(blue colour only on the right hand side)

The proposed model in this paper however has very few limitations and performs eventually better than its counterpart. It requires very limited hardware components, the most important one being the camera. Since most of the devices comes with an integrated camera, there is no need of purchasing any other sensors to perform the task. This minimises the cost of deploying our model, *meanwhile maximising the responsiveness across* multiple devices and scale *up when there is a need in the future.*

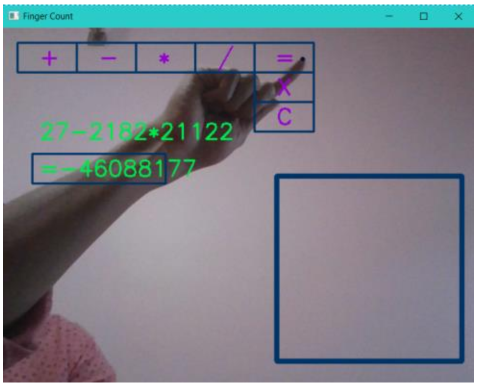

Fig.2 Shows the sample computation by our model, complex expressions could be written hassle-free and calculations were performed with ease.

The paper organization is as follows: Section II highlights the various aspects of algorithms that are currently being used to detect the hand gestures; Section III introduces the training method and the methodology based on which the model has been built; Section IV discusses the introduction & illustration of the software that are to be installed and tuned. Also depicted is a breakdown of the software approaches that were used while building this model; Section V presents the challenges faced while performing the hand gesture based recognition systems, Summary and conclusion

II. LITERATURE SURVEY

The HSV mode of segmentation was one important method in gesture detection. In this method, Image segmentation uses HSV color space model rather than RGB color space to determine the color of human skin. This algorithm gives better result for background separation and region boundary but it failed to detect the object which has similar to that as of skin. The approach has been taken by our team but was used in different context. Our team used this same technique to segment a colored ball , whose movement was tracked to take gesture input. We were convenient in using that, as we were able to use the ball, like a pen to write. The ball was just hovered around and when the ball goes out of frame/hidden, the tracking was stopped and the value of the digit (which was drawn on the screen) was predicted by the model which was trained to predict with the help of MNIST dataset. [1] Alternately there is a method using convexHull (which is much similar to our method) in order to detect the gestures.In this method the final decision is made either by checking the ratio between the length and the width of the bounding box against four intervals of values, or by measuring the distance between the upper corners and the hand. [2] In the model proposed by Sagar P.More and Abdul Sattar, they suggested to reduce the dimension of the feature vectors, to discover more distinctive features and then apply dimensionality reduction to the existing descriptors feature vectors so that information could be extracted from them. But more light intensity exposure on hand gesture images affected the result and lead to decreased accuracy. There are more algorithms to figure out gestures but for designing hand gesture recognition system high precision segmentation algorithms are required. Not to forget that the rate of recognition is trade-off with the time rate. Therefore, the emphasis should be focused on to improving the time rate[6].

III. METHODOLOGY

The usage of both left and right was necessary to do the calculations and we had designed it in such a way that left hand is for operators and the right hand is for numerical. The principle behind this design was that, it has been proven that right hand is coordinated by the left hemisphere of the brain which is capable of dealing with large numbers and is good at math. To improve the user experience and to ease the way in which human interacts with the system, we have carefully designed the positioning of the toolbars in the screen. A Convex object is one with no interior angles greater than 180 degrees[5]. Hull means the exterior or the shape of the object. Therefore, the Convex Hull of a shape or a group of points is a tight fitting convex boundary around the points or the shape. The convex hull is based on the Sklansky algorithm, though it was proven to be incorrect it is still in use and gives correct result in most of the times.

In order to segment foreground and background, the initial frames (200 in our case) were aggregated and the running average is computed. So for the initial "n" frames, any part of the body shouldn't be brought into the coverage of the webcam. *When "n" frames are captured, we get it* *notified in the screen and so we start with the gestures.*

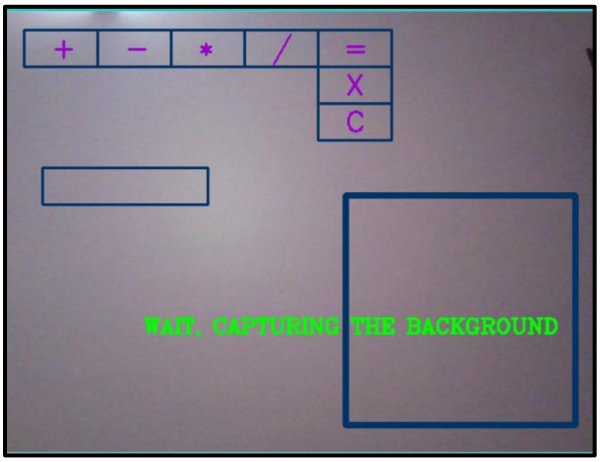

Fig 3,initial background capture for first 200 frames, to segment foreground from background

Now every frame is subtracted from the background (which we captured at the beginning) to segment the hand from the image. Here the hand was not part of the frame when we captured the background, so when we perform background subtraction we, get the grayscale values of the hand segment while the rest of the pixels are 0(black). Then thresholding is applied to perform segmentation on binary level where the hand pixels are 1(white) and the rest of the pixels are 0(black) as shown in figure 4. We apply thresholding to perform contour detection which helps us to obtain the hand segment and to create a Convex Hull around the hand. Assuming that the hand covers most pixels of the frame, we assign the hand segment to the max of contours, with the key being the area of the contours. Before applying contour detection, morphological close is applied to the image to remove the white noise present(if any). Gaussian Blur is also applied before segmentation to get better results.

Fig 4 Thresholded image of the hand after segmentation

(A Convex object is one with no interior angles greater than 180 degrees. Hull means the exterior or the shape of the object. Therefore, the Convex Hull of a shape or a group of points is a tight-fitting convex boundary around the points or the shape. The convex hull is based on the Sklansky algorithm, though it was proven to be incorrect it is still in use and gives correct result in most of the times). A Convex Hull is created around the hand segment to understand the number depicted by the hand gesture. With the convex hull we can now find out the number of fingers that are being extended and the number of fingers that are being folded. The region that are not covered by the hand inside the convex hull are the convex defects, which are utilised to count the no of fingers that are extended. Angle between fingers(defects) are calculated by using cosine rule as follows:

$$A = \cos^{-1}\left(\frac{b^2 + c^2 - a^2}{2bc}\right)$$

The "start', 'end', 'far' points of a defect are identified and used for the calculation of the sides of the triangle formed by the defect (a, b, c).

$a = \sqrt{((end[0] - start[0]) ** 2 + (end[1] - start[1]) ** 2)}$

$b = \sqrt{((far[0] - start[0]) ** 2 + (far[1] - start[1]) ** 2)}$

$c = \sqrt{((end[0] - far[0]) ** 2 + (end[1] - far[1]) ** 2)}$

The angle between the fingers cannot be greater than 40, so this condition is applied to filter the defects that are formed between the two fingers and not between a finger and the hand. Now we can count the number of defects (blue dots in the image shown below) and add one to it to find the no of extended fingers. Thumb is detected by using the condition that the angle of the defect should be greater than 50 and less than 100. In order to avoid counting some unrelated defects, we have added an additional condition that the distance/depth of the defect should be less than 3. Now by knowing the no of extended fingers and whether the thumb is present or not, we can *identify the number depicted by the respective hand gesture. I have set up a buffer of 50 frames, until the next input is taken, to have a smoother user experience.*

Now if a user wants to add an operator to the equation, then the user has to remove his/her hand from the ROI_N and hover over the operators in the display (ROI_O) to choose the respective operator. It is necessary to remove the hand from ROI_N because higher preference is given to ROI_N, so the operator won't get selected otherwise. The same process is applied to ROI_O as that applied to the ROI_N till the formation of convex hull. The next step is to find the coordinates of the top most point of the convex hull and to find out within which operator's box the top point lies. The respective operator in the corresponding box will be added to the equation. And finally, when the "=" is selected, the result of the equation is computed with the help of eval() function and the solution is displayed.

IV. SOFTWARE IMPLEMENTATION

The model had been programmed in the python programming language with the help of the following packages.

i)numpy
ii)openCV
iii)matplotlib
iv)scikit learn
v) math

Jupyter notebook was the IDE used here since usage of google colab to turn on the webcam was buggy, and often lead to freezing of the screen. To perform a hardware test(here, checking if the camera is properly working) we can use the isOpened( )method that comes inbuilt with openCV. An object with reference to videoCapture(camera number) is declared and the isOpened boolean valued function returns the working of the camera. The camera that was used for the video input is the integrated camera that comes pre-built with the laptop. This Web camera provides video capturing with a maximum resolution of 640 x 480 up to 30 frames/s. The experiments are implemented with LED(Light Emitting Diode) lighting conditions. Opencv provides functions for contour detection, contour drawing, convexity defects and identifying the convexhull as well. The input frame is made to undergo a Gaussian blur with kernel size of 5. This is done to ensure that spurious high-frequency information does not appear in the down sampled image.

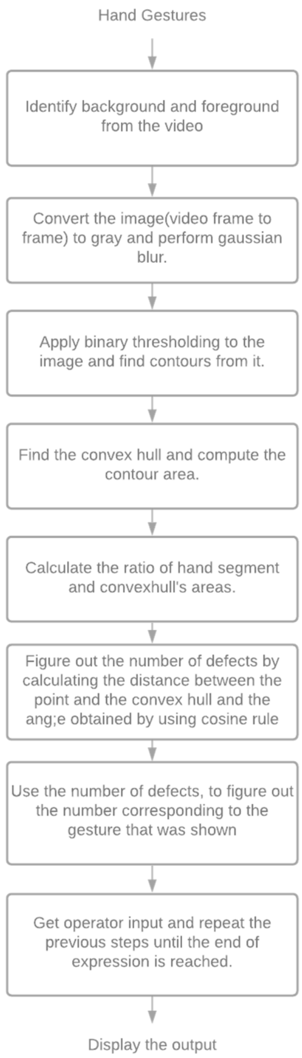

Fig.5 The steps involved in gesture detection and numerical computation.

V. CONCLUSION

All this requires analyzing the entire image/video in the browser window. This project has been carried out under the following environment. A 2.40 GHz intel® core™ processor , 8 GB RAM , Windows based image processing software & Jupyter notebook IDE is used to analyse images and videos with a frame rate of 30 frames per second. The aim of this project to develop a Gesture based calculator was achieved. We tried our best possible ways to enhance the working of the model but there were some cases in which the model was not able to provide accurate results. It was determined that contour is a very important feature and the gesture is interpreted by the *computer only based on the defects found in hands. The number of defects is taken into account to interpret the number and the model was unable to distinguish some gestures when the lighting was changed. On few instances, if there was an external interference (especially of those objects with traits similar to those of the skin tone) into the frame after the initial background subtraction, the model had lost its tracking. The model was able to compute the given expression in no time and the result was clearly displayed in the output box with accurate decimal precision.*

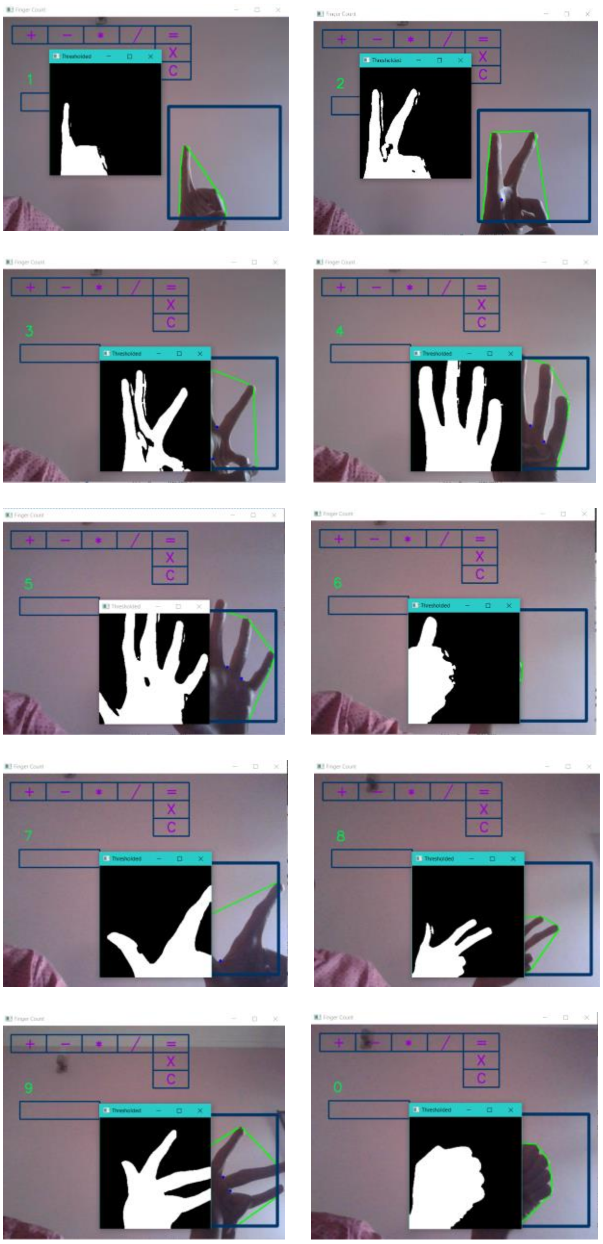

Fig.6 Various gestures for different numerical input.

VI. ACKNOWLEDGEMENT

We would like to express our gratitude to our supervisor Dr.Anisha M.Lal for her guidance, advice and constant support throughout our thesis work. We would also like to thank Dr.Dheeba.J for assisting us with the publication works and motivating us to get involved in research projects.

VII. REFERENCES

[1] Ruchi Manish Gurav,Premanand K. Kadbe "Image Processing Algorithm for Appearance-Based Gesture Recognition", 2015 International Conference on Industrial Instrumentation and Control (ICIC) College of Engineering Pune, India. May 28-30, 2015

[2] Sagar P.More and Prof. Abdul Sattar, "Hand Gesture Recognition System Using Image Processing", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016

[3] Ebrahim Emami, Mahmood Fathy "Object Tracking Using Improved CAMShift algorithm Combined with Motion Segmentation",IEEE 2011 7th Iranian Conference on Machine Vision and Image Processing ,2011

*[4]*Soowoong Kim, Jae-Young Sim, And Seungjoon Yang "Vision-Based Cleaning Area Control For Cleaning Robots", IEEE Vol. 58, No. 2, May 2012

*[5]*M.M. Hasan, P.K. Mishra, "Hand Gesture Modeling and Recognition using Geometric Features: A Review", Canadian Journal on Image Processing and Computer Vision, vol. 3, pp. 12-26, March 2012.

*[6]*Jayesh S. Sonkusare,Nilkanth. B. Chopade,Ravindra Sor,Sunil.L. Tade," A Review on Hand Gesture Recognition System", 2015 International Conference on Computing Communication Control and Automation