As per UGC guidelines an electronic bar code is provided to seure your paper

International Journal for Modern Trends in Science and Technology, 7(01): 23-26, 2021
Copyright © 2021 International Journal for Modern Trends in Science and Technology
ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0701005
Available online at: http://www.ijmtst.com/vol7issue01.html

Authenticating user while Performing Transaction at ATM/POS using Hand Gestures

Rahul Gupta[1] | Keshav Choudhary[1] | Sagar Kashyap[1] | Vishisht Chhabra[1] | Jyoti Verma[2]

[1]Student, Department of Computer Science and Engineering, Dr. Akhilesh Das Gupta Institute of Technology & Management, Delhi, India
[2]Assistant Professor, Department of Computer Science and Engineering, Dr. Akhilesh Das Gupta Institute of Technology & Management, Delhi, India

To Cite this Article
Rahul Gupta, Keshav Choudhary, Sagar Kashyap, Vishisht Chhabra and Jyoti Verma, "Authenticating user while Performing Transaction at ATM/POS using Hand Gestures", *International Journal for Modern Trends in Science and Technology*, Vol. 07, Issue 01, January 2021, pp.-23-26.

Article Info
Received on 22-November-2020, Revised on 18-December-2020, Accepted on 22-December-2020, Published on 28-December-2020.

ABSTRACT

This electronic document defines the use of Vision-based technology for authentication at ATM/POS while performing a transaction.
It makes use of Hand Gestures which are conveyed through finger position and shape of a palm, which helps in entering the ATM pin through gestures of the hand.
The hand gestures are being detected using OpenCV, python 3.7.
A High-uality camera and well-lit background are required for proper detection of hand position.
This Paper will propose to develop a hand gesture recognition system for safely performing a transaction at ATM/POS.

KEYWORDS: *Covex Hull, Contour, OpenCV, Convexity Defects, Cosine Theorem, HLS (Hue Lightness, Saturation)*

I. Introduction

Vision-based technology such as image recognition, face recognition, smile detection, text recognition is an important part of Human-Computer interaction. Owing to the rapid development of Human-Comp interaction new methods such as speech recognition and gesture recognition have received great attention in this field. Gestures are the movement of body parts such as the head or hand to express emotions or body language.

One such gesture - Hand gestures are conveyed through finger position, the center of the palm, and shape of the hand. They offer an inspiring field of research as a means of communication and neutral interaction. With the evolution of open-source libraries detection of hand-gestures can be used in a wide range of applications such as gaming, sign language, home automation, personal computer, and tablet. Authenticating users while performing a transaction at ATM/POS securely is also an application of hand gestures detection.

In this paper, we would propose, how to develop a hand gesture recognition system for safely performing a transaction at ATM/POS using OpenCV and Python 3.7. Our proposed approach is divided into three stages:

i. Preparing a binary mask
ii. Computing contour and its convex hull
iii. Detecting fingertips in the first step, creating binary mask for a visusl image to calculate hand lines.

II. Working

i. Preparing the binary mask

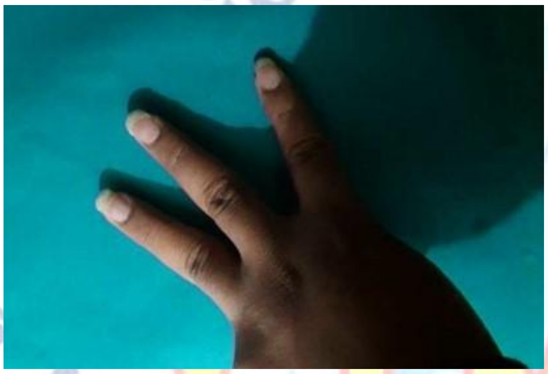


Fig 1.0: Picture of Hand

We need to create a binary mask of hand to compute the contour. We classify the image by skin color using the in range function. We convert our frame to Hue Lightness Saturation (HLS) color space. The hue channel encodes the actual color information correctly.

(Shown in Figure 1.0)

For the sake of those, we only need to find the right range of Hue skin value and we only need to adjust the Saturation and Lightness values.

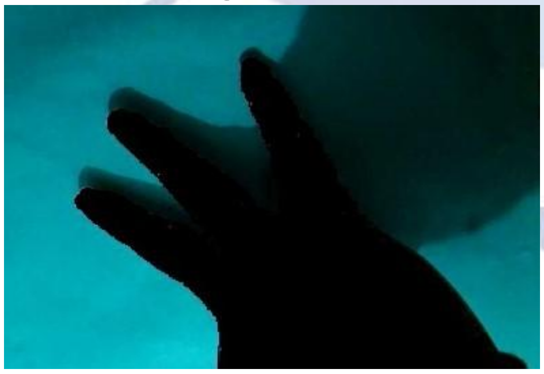


Figure 1.1: Binary Mask of Hand

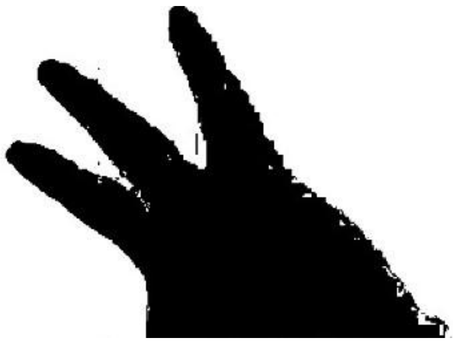


Figure 1.2: Region of Interest

It converts our frames (shown in Fig. 1.1), available in BGR format, as we read in a file or scan in OpenCV, to the HLS color space (Hue Lightness, Saturation) (shown in Fig. 1.2). The hue channel includes real color details [IV].

For example, if we get a Hue range of 0 to 40 degrees then the filling distance will be 5% to 70% using a simple color key. In OpenCV the hue change will be 0 to 180 degrees instead of 0 to 360degrees, thus we will filtering the image with Hue in the range of 0 to 15 then divide by 2. Removing noise such as single pixels or small spaces between the fingers improves the hand mask slightly by smoothing out the blurring and boundary function and subsequently obtaining an accurate binary mask again [IV].

ii. Computing the contour and its convex hull

We will find the contour in the mask by OpenCV and return the largest contour, if the part is not working properly and is still noisy (shown in Figure 2.0 and Figure 2.1)[I]. Now we have to find the line to use the actual algorithm to find the fingers and the number of fingers displayed on the camera. The Point of the ships is pointed by red circles. Next we need to make sure that only one finger has a set of one point and take the middle point of each set.

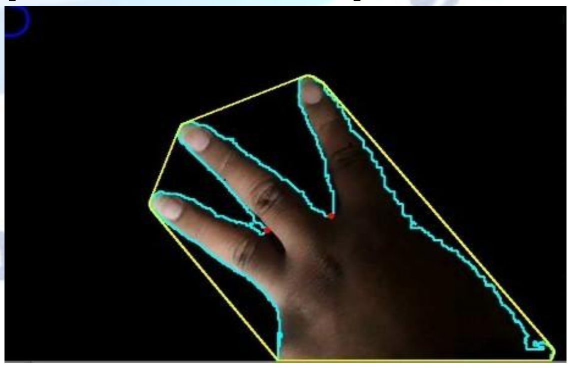


Fig 2.0: Convex Hull

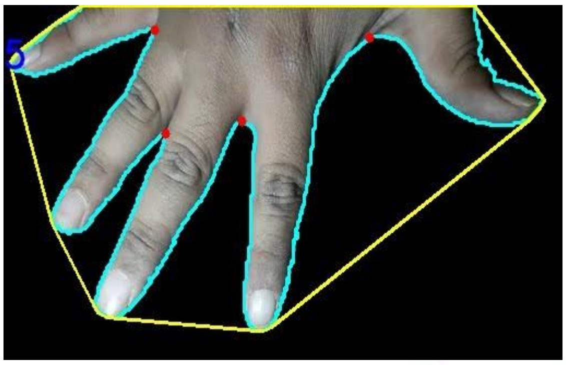


Fig 2.1: Convexity Hull

In compiling using CV partition, we will feed the point of the line below the space of the coves and assign a return function, which will calculate two points and determine whether they belong to one group. If the distance of points is below the "maxDist" threshold, then they are assigned to the same group. The resulting polygons should look much clearer than before.

iii. Detecting Fingertips

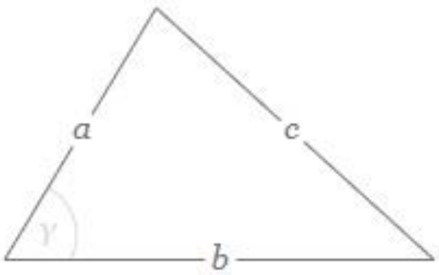
OpenCV now calculates the indulgence of a hand-held combination in our new polygon[III]. This will give us a region with features of our line from the beginning to the end of the disability point. The angle between two fingers is small to be angle alpha then need to change. We use another angle beta instead of which work as well. Therefore we will transfer the data as follows. The defect regions are returned to the array of vectors., each boat is assigned to a nearby neighbor point and those points are not considered which do not have two neighbors. of each vertex with the lower insertion of cosines into them.

iv. Fingers to Numbers

After the convex Hull help for hand is being created. Three Points are being found:

1 Start point of convex hull
2 End point of convex hull
3 Farthest point of convex hull

For each convexity defect in a convex hull which includes the farthest point, a triangle is being made whose vertices are start point end point and the farthest point of convex Hull.



$$c = \sqrt{a^2+b^2-2ab\cos\gamma}$$

Fig 3.0: Cosine Theorem

Making the use of cosine theorem, the angle for each convexity defect is being found that includes the farthest point.

The number of conductivity defects whose angle is less than 90 degrees is being calculated, which is a power output except for input digit 1.

If no convexity defect is being found having angle less than 90 degrees we consider that case for the output for input digit 1.

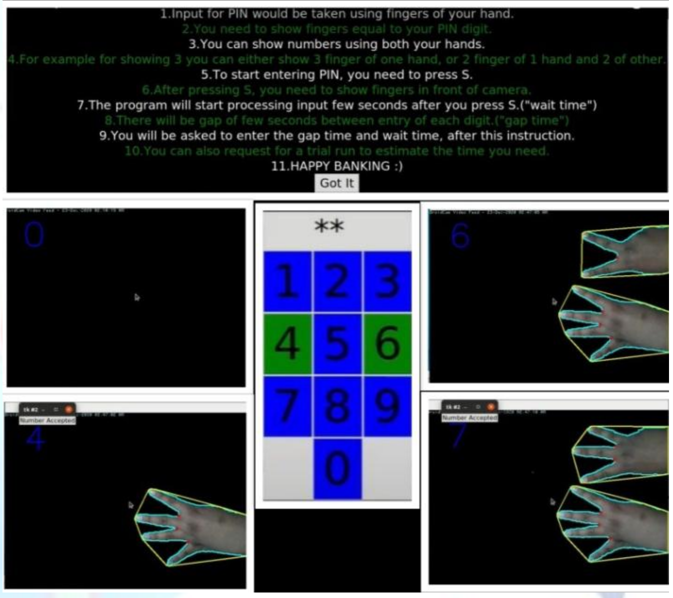

1.Input for PIN would be taken using fingers of your hand.
2.You need to show fingers equal to your PIN digit.
3.You can show numbers using both your hands.
4.For example for showing 3 you can either show 3 finger of one hand, or 2 finger of 1 hand and 2 of other.
5.To start entering PIN, you need to press S.
6.After pressing S, you need to show fingers in front of camera.
7.The program will start processing input few seconds after you press S.("wait time")
8.There will be gap of few seconds between entry of each digit.("gap time")
9.You will be asked to enter the gap time and wait time, after this instruction.
10.You can also request for a trial run to estimate the time you need.
11.HAPPY BANKING :)

Fig 3.1:Inserting PIN

And then the input pin is being shown as seen in figure 3.1.

III. Research Challenges and Limitations

From previous sections, much easy to identify research gaps are identified, that is many foci on enhancing frameworks instead of a robust framework which becomes the biggest challenge that is overcoming the most common issues with fewer limitations and gives accurate and reliable results. We used a simple method, which is to use image processing techniques in the OpenCV library, which monitors the use of time for real-time processing. This has limitations such as

i. Background Issues – Background objects along with the hand image
ii. Illumination Variations – Brightness and contrast of the hand image provided
iii. Distance Limit – Close to camera hand images or lengthy distance images causes unnecessary

problems

iv. Multi-object or Multi-gesture Processing – Multiple objects or more than one hand in an image causes a system identification problem in contour for finger detection.[II]

IV. ACKNOWLEDGEMENT

We would like to extend our special thanks to our college for giving us this excellent opportunity to explore on this topic (Assuring the user while signing The ATM / POS securely), which also helped us in learning a lot of new things.

We would also thank our college, teachers, parents and friends who supported us so much in completing this work in a timely manner.

V. REFERENCES

[1] https://docs.opencv.org/master/d1/dfb/intro.html
[2] https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html
[3] https://docs.opencv.org/master/d7/d1d/tutorial_hull.html
[4] https://docs.opencv.org/master/df/d9d/tutorial_py_colorspaces.html
[5] D. Bagdanov, A. Del Bimbo, L. Seidenari, and L. Usai, “Real-time hand status recognition from RGB-D imagery,” in Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12), pp. 2456–2459, November 2012.
[6] Y. Wang, C. Yang, X. Wu, S. Xu, and H. Li, “Kinect based dynamic hand gesture recognition algorithm research,” in Proceedings of the 4th International Conference on Intelligent HumanMachine Systems and Cybernetics (IHMSC '12), pp. 274–279, August 2012.
[7] Marium A, Rao D, Crasta DR, Acharya K, D’Souza R (2017) Hand gesture recognition using webcam. Am J Intell Syst 7(3):90–94