RESEARCH ARTICLE OPEN ACCESS

Hand Gesture Recognition and Appliance Control Using Transfer Learning

Srinivasa Rao K , Gundam Bhavani ,Tirumalasetty Veena Madhuri ,Pagolu Sriharika

Department of Electronics and Communication Engineering, Andhra Loyola Institute of Engineering and Technology, Vijayawada,India

ABSTRACT

The computer is becoming more important in our daily life with the development of ubiquitous computing. Computer applications require interaction between human and computer.A hand Gesture is basically the movement, position or posture of hand used extensively in our daily lives as part of non-verbal communication. A lot of research is being carried out to classify hand gestures in videos as well as images for various applications. The physical movement of the human hand produces gestures, and hand gesture recognition leads to the advancement in automated vehicle movement system. In this paper, the human hand gestures are detected and recognized using transfer learning approachin python.This process flow consists of background subtraction,hand ROI segmentation,Contours detection and fingers recognition using transfer learning CNN trained model.The hand region of image is segmented from whole image using thresholding technique.The system has two major advantages. First, it is highly modularized, and each of these steps is capsuled from others; second, the contour detection of hand as well as gesture recognition is an add-on layer, which can be easily transplanted to other applications.The techniques used for image processing are hand gesture detection, pattern recognition, thresholding, contour detection.Using OpenCV, which provides a library collection of functions for different image processing techniques, these input images can be processed and corresponding key strokes will be generated. The proposed hand gesture detection and recognition methodology using deep learning CNN approach with enhancement technique stated in this paper achieves high performance.

Keywords— CNNclassifier,FaceRecognition,FaceDetection,Threshold, Segmentation

Date of Submission: 05-07-2021 Date of Acceptance: 18-07-2021

I. INTRODUCTION

Hand gesture recognition provides an intelligent and natural way of human computer interaction (HCI). Its applications range from medical rehabilitation to consumer electronics control (e.g. mobile phone). In order to distinguish hand gestures, various kinds of sensing techniques are utilized to obtain signals for pattern recognition.At present, human–machine interaction is very important for operating the machines in a remote manner by the commands which are received from humans. In this regard, gestures are playing an important role in operating the machine at a distant mode (Yasukochi et al. 2008). The machines capture the gestures from the human and recognize it for operating the machines. The gestures are different types of modes as static and dynamic. The static gestures do not change their position, while the machine is operated, and the dynamic gestures change their positions during the machine is operated (Elmezain et al. 2010; Mitra and Acharya 2007).

Hence, the identifification or recognition of dynamic gestures is very important than the static gestures (Yrk et al. 2006; Tauseef et al. 2009). Initially, the camera, which is connected with machine, captures the gestures which are generated by humans.The background of the detected gestures is removed, and the foreground of the gesture is captured.After background subtraction segmentation of hand ROI image is separated from given iamge.Here segementation technique used is thresholding method,which has simple preprocessing algorithm.After thresholiding we are going to extract the contours.After that we are going to find convex hull,angle between fingers.Based on these calculations we are going to count fingers.

The automotive sectors and many consumer electronics division use the gesture-based machine operating system without any human interaction. Besides the static and dynamic gestures, the gestures of human are also classified into online and offlfline gestures. The offlfline gestures operate the icons on the machine, and they are not able to alter the position of the items in the menu or system. The online gestures operate the icons in the machine to different positions or inclinations (Yao and Fu 1935; Liu et al. 1898). The online gestures are very much useful in realtime machine operating systems than the offlfline gestures.Park et al. (2012) and Ren et al. (2013) used Naı¨ve Bayes classififier and support vector machine (SVM) methodologies for gesture recognition. These methods did not support large number of training dataset, and it also required high number of training samples. This drawback is eliminated by proposing transfer learning based CNN classififier in this paper. It does not require high number of samples in training mode, and the complexity level of this algorithm is low. The novelty of this proposed work is to implement deep learning algorithm in hand gesture recognition system with novel segmentation technique.

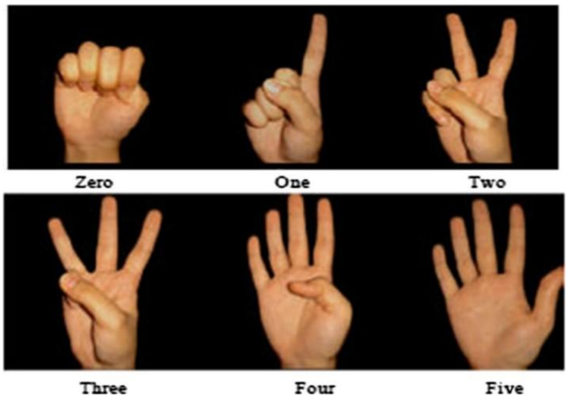


Fig. 1 Different hand gesture postures (Kawulok et al. 2012)

The paper is organized as follows: firstly deals the conventional methods of hand gesture recognition, proposes CNN classifification-based hand gesture recognition, and discusses the simulation results of proposed hand gesture recognition system using Python programming language. At the end one application using hand gesture recognition concept, concludes the paper.

In this study, designing of the hand gesture recognition is one of the complicated job that involves two major problem. Firstly is the detection of hand. User hand is detected by using webcam in real-time video. The problem would be the unstable brightness, noise, poor resolution and contrast. The detected hand in the video are recognized to identify the gestures. At this stage, the process involves are the segmentation and edge detection or background subtraction. According to Karishma [2], with various information of image like color, hand posture and shape based (shape of hand) in a real time would affect the recognition of gestures. Another problem is to create the sign that is suitable to be used for one hand in a time. The extraction of hand need to be followed to determine each number and sign used. According to Amiraj [3], the extraction of hand involves the contour and convexity defects. Convexity defect gives an issue on how to calculate the depth of the defects. Some of the defects have far greater depth than others so to identify the depth would include some equations.

The list of objectives that will need to be achieve for this project: (1) to establish a complete system for detecting, recognizing and interpreting hand gesture recognition through computer vision using Python and OpenCV, and (2) to create the numbers and sign languages of hand gesture shown in the system that will meets the name of the project.

II. LITERATURE SURVEY

Zuocai Wang et al. (2018) proposed hand gesture recognition system using particle fifiltering approach. The authors applied this fifiltering approach on hand gesture images with same background. The authors obtained 92.1% of sensitivity, 84.7% of specifificity and 90.6% of accuracy. Suguna and Neethu (2017) extracted shape features from hand gesture image for the classifification of hand gesture images into various classes. Then, these extracted features were trained and classified using k-means clustering algorithm. Marium et al. (2017) proposed hand gesture recognition system using convexity algorithm approach.The authors applied this fifiltering approach on hand gesture images with same background. The authors obtained 90.7% of sensitivity, 82.1% of specifificity and 87.5% of accuracy. The main limitation of this approach is that the proposed algorithm produced optimum results if the background of the hand gesture image is static.Ashfaq and Khurshid (2016) used Gabor fifiltering approach for converting the spatial domain format hand gesture image into multi-class domain format image. Then, the authors applied both Bayesian and Naı¨ve Bayes classifier on Gabor transformed hand gesture image in order to classify the test hand gesture image into different classes.

The authors obtained high level of classifification accuracy on Naı¨ve Bayes classifier than the Bayesian classifification methodology due to its simple architecture pattern.

Rahman and Afrin (2013) used support vector machine (SVM) classifification approach for classifying the hand gesture images into various classes. The authors achieve 89.6% of sensitivity, 79.9% of specifificity and 85.7% of accuracy. The error rate was high in this method, and this is not suitable for fast moving background and foreground object images. Rao et al. (2009) developed hand gesture recognition system using hidden Markov model.

The authors constructed Markov model for foreground fingers in hand gesture image. This Markov model was used in both training and testing modes of binary classification approach. The authors produced 90.1% of sensitivity, 82.6% of specificity and 90.6% of accuracy. The classification time is high in this methodology as the mail limitation.

The following points are limitations of the conventionalmethods for gesture recognition.

- Conventional gesture recognition method used SVM and Naı¨ve Bayes classifier, which required high number of training samples for gesture patternrecognition.
- The complexities of these algorithms are quite high innature.

III. PROPOSED METHODOLOGY

In this paper, the human hand gestures are detected and recognized using CNN classification approach. This process flflow consists of hand ROI segmentation using mask image, fifingers segmentation, normalization of segmented finger image and fifinger recognition using CNN classifier. Figure 2 shows the proposed flow of hand gesture recognition system

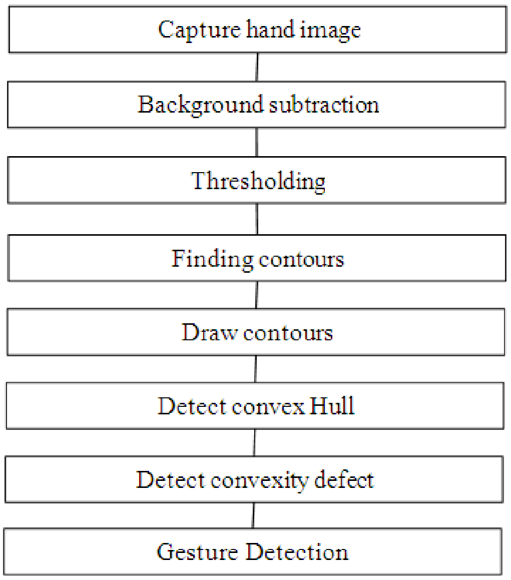


Fig:2 Process of hand gesture recognition

IV. IMPLEMENTATION

The proposed system was coded by Python language,OS Module of Python and the OpenCV library. And also here we used pre trained transfer learning CNN based model for hand gesture recognition.Transfer learning is a deep learning method where a model developed for a task is reused as the starting point for a model on a second task.That model is tested and trained by using deep learning convolutional neural networks used for hand detection and recognition.These model is designed by using Keras and Tensarflow tools.Keras is used for frontend purpose to train a model in python where tensorflow is used as backend tool.

The general structure for any system to recognize the hand gestures can be explained as shown in Figure 2. Our system contains five steps, which are detailed.

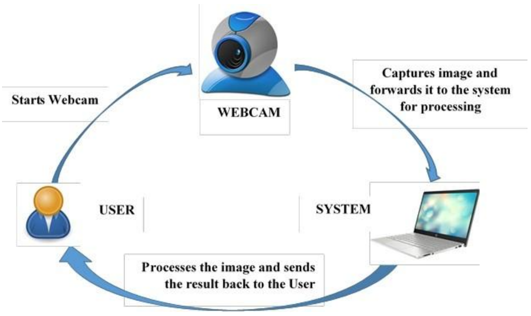

Starts Webcam
WEBCAM
Captures image and forwards it to the system for processing
USER
SYSTEM
Processes the image and sends the result back to the User

Fig2. General structure of the hand gestures recognitionsystem

4.1. Image Capturing

In this step used a webcam to acquire the RGB image (frame by frame) and based on only bare hand withoutglove.

4.2. Pre-Processing

In this step in order to minimize the computation time we took only the important area instead of the whole frame from the video stream and this is called Region Of Interest (ROI).

In image processing prefers to convert the color images into a grayscale images to increase the processing and after complete the processing can restore the images to its original color space, therefore, we convert region of interest into a grayscale image. Then blurring the (ROI) by Gaussian blur to reduce the objects that have high frequency but not the target. Notice that in this step the algorithm will fail if there is any vibration for the camera.

4.3. Hand Region Segmentation

This step is important in any system to hand gestures recognition and help in enhancing the performance of the system by removing the unwanted data in the video stream. In general, there are two methods to detect the hand in image, the first method depends on Skin-Color, this method is simple but effected by the light conditions in the environment and the nature of background. The second method does not depend on Skin-Color but on shape of hand and benefit from the principle of convexity in detection of the hand. The shape of hand is very important feature in the systems of recognition the hand gesture.

There are several methods or techniques used to extract the hand region from the image can be summarized as:

a) Edge-Detection.
b) RGB values because of the values of RGB for hand differentfrom the background of the image.
c) Subtraction of background

In this paper used background subtraction technique to separate the hand from the background. This technique needs to determine the background that can be obtained by benefit from a running average principle. The background is computing from made the system focus on a certain scene for at least 30 frames and during that calculating the running average for the current frame and all previous frames using the following equation: dst(x,y)=(1−a).dst(x,y)+a.src(x,y)
where, src (x,y) is a source image may be one or three channels and 8-bits or 32-bits floating point, dst (x,y) is destination image contains the same channels in the source image and 32-bits or 64-bits floating point. Finally, alpha is a weight of the source image and can be considered as threshold to determine the time for computing the running average over the frames. After determining the background, we put the hand front the camera, then compute the absolute difference between the background that calculates by using the running average and the current frame that contains the hand as a foreground object. Thus find the background and then compute the difference all that called subtraction background. The next step is thresholds the image that output from background subtraction and the result will be only hand with white color and the rest image with black color. The threshold process is important and must be done before the contours finding a process to achieve high accuracy.

Mathematically can represent the threshold principle as follows:

$$f(x) = \begin{cases} 1, \; if \;\; x \geq threshold \\ 0, \; if \;\; x < threshold \end{cases}$$

where, f(x) is the intensity of the pixel.
All the processes above called Motion-Detection. Figure 3 shown the output of Hand region segmentation process. Finally, perform a chain of morphological processes such as erosions and dilations to remove any small regions of noise.

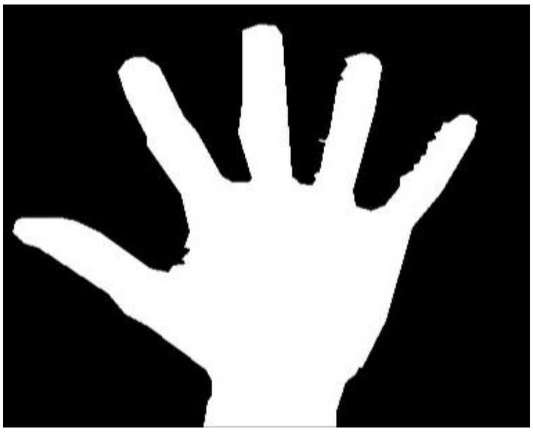


Figure 3. Output of hand region segmentation process

4.4. Contour-Extraction

The contour can be defined as object’s boundary or outline (hand in our case) that be located in the image. In other words, the contour is a curve connecting points that have the like color value and is a very important in shape analysis, objects detection and recognition process.

4.5. Features Extraction and Recognition

Now we turn to the second part of the research, which is how we determine the number of fingers. From the number of fingers can recognize the hand gestures, and for performing this task used Convex Hull to locate the extreme points (top, bottom, left and right). The convex hull is the group of points that surrounding the region of hand as shown in Figure 4. Here, we must clarify the principle of the

4.5.1 Convex Hull method

Convex Set, which means all lines between any two points within hull are entirely within it. From extreme points can compute the palm's center, Figure 5 shown that

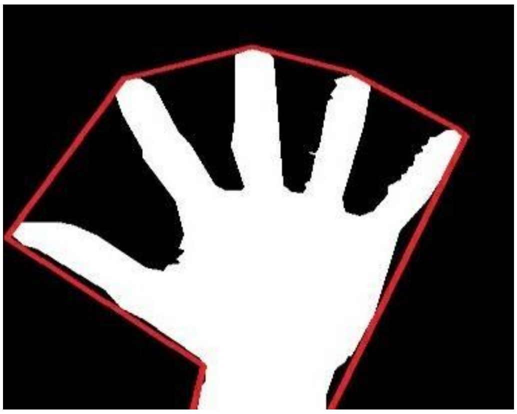


Figure 4. Convex hull

This method was inspired by an online tutorial by ifheqhar. The convex hull method takes the outline of a shape, and identifies the convex and concave (defect) points along the outline. These points provide us with a rough idea of the shape of the object. In the case of a hand, it should have 5 convex points (one for each finger) and 4 defects (one between two adjacent fingers).

Using this method, we can identify the number of fingers the user is showing to us by the number of convex and defect points. For example, if there are 2 convex and 2 defect points, it is likely that the user is showing us 2 ingers. In our implementation, we used OpenCV functions such as findContours(), convexHull() and convexityDefects() to obtain the convex hull and defect points. Below is a sample output of the centroid and number of defects for a given detection.

The next step is to draw a circle about the fingers, its center point is the center of the palm and the seventy percent from the length of maximum Euclidean distance between the palm's center and extreme points represent the radius. Any deviation of the object from this hull can be considered as convexity defect.

We can visualize it using an image. We draw a line joining startpoint and end point, then draw a circle at the farthest point.

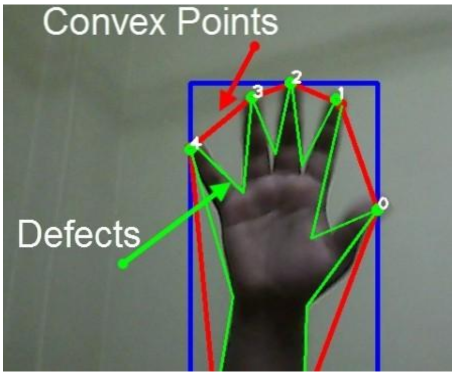


Figure 5.Convexity defects

We have to apply cosine rule to find angle for all defects between the fingers.In trigonometry, the law of cosines relates the lengths of the sides of a triangle to the cosine of one of its angles. The law of cosines states where γ denotes the angle contained between sides of lengths a and b and opposite the side of length c.

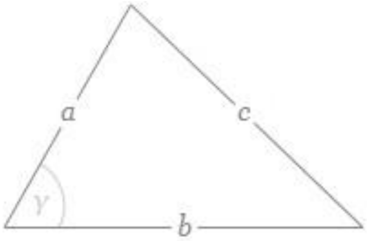


Figure 6.Cosine rule triangle

Formula

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

For finding gamma this formula is used:

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

In the figure 7 a,b,c are three sides of a triangle and gamma is the angle and this angle always should have less than 90 degrees value.If gamma is less than 90 degree or pi/2 we consider it as a finger.

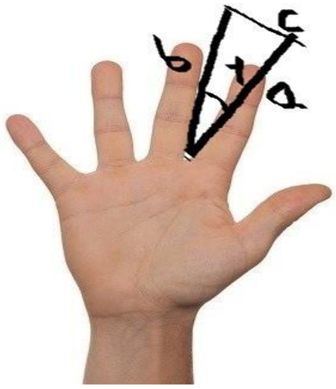


Figure 7.Finding angle between fingers

As we read earlier, if gamma is less than 90 degree we treated it as a finger.After knowing gamma we just draw circle with in approximate distance to farthest point.And after we just simple put text in images we represent fingercounts (cnt).

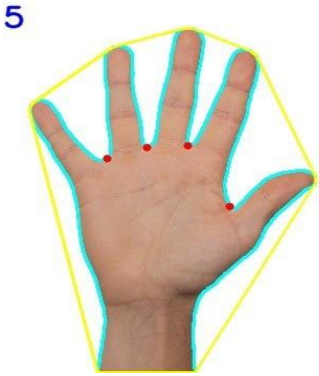


Figure 8 Counting fingers

The process of distinguishing the hand gesture is a dynamic process. After performing the required instruction from the gesture, return to the first step to take another image to be processed and so on.

V. APPLICATIONS

Coming to applications of hand gesture recognition method now a days we are using in several applications in real time like in medical applications advanced robotics systems with gesture recognition can be placed in hospitals or homes to recognize and treat life threatening conditions like hear attacks.In alternative computer interfaces,entertainment applications,automation systems,an easier life for disabled.In this paper we designed one application based on hand gesture recognition that is media player controlling by using hand gestures.In this papaer we designed a media player system that has been controlled by various hand gestures consists of play, and pause, Full screen, and stop, increase volume, and decrease volume features.In this paper we have discussed a low cost system which uses dynamic hand gesture recognition technique to control the VLC media player.

This application contains a central computation module which segments the foreground part **of** the frame using skin detection and approximate median technique. This hand gesture recognition technique introduces a new, natural way to interact with computers.This media player controller process includes capturing hand image,convert it into hsv,tracking hand on color basis,creating mask on the basis of color and filter actual color,invert pixel value and then enhancing the result for the better result,finding contours for specific colored object,finding max area contour and draw it on live feed,finding convexity defect for counting values and apply cosine method,binding hand gestures with keyboard keys these are the steps involved in media controller using hand gestures.

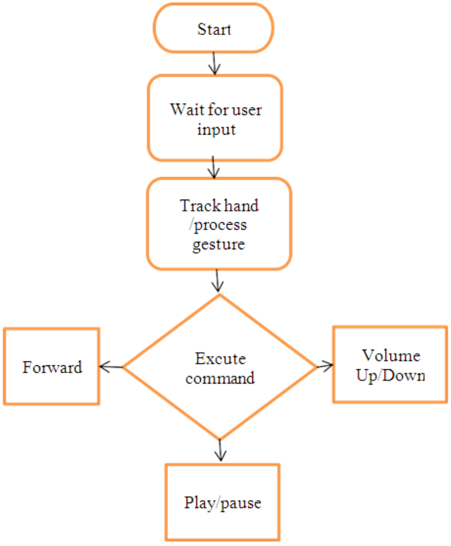


Figure 5.1 process of controlling media player by handgestures

After implementing these steps we got results as shown inbelow figures.

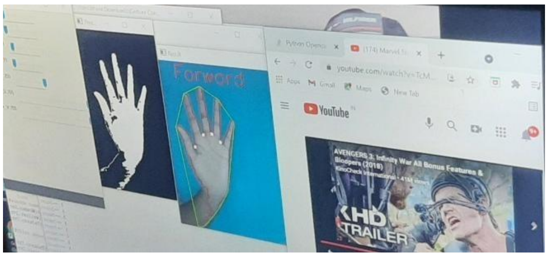


Figure 5.2 Forward the video

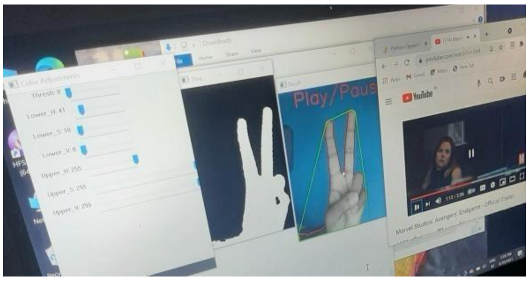


Figure 5.3 Play/pause

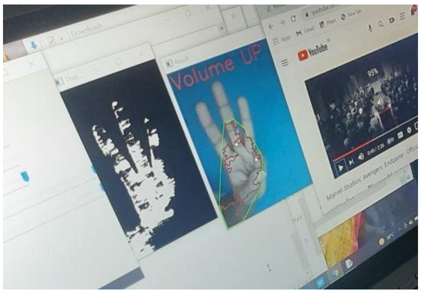


Figure 5.4 Volume up

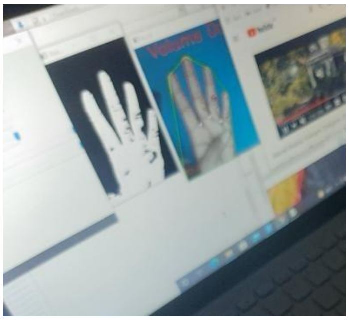


Figure 5.5 Volume down

Above figures are the results of controlling the media palyer by using hand gestures.

VI. RESULTS

The proposed hand gesture detection and recognition methodology is simulated using Python version 3.6 opensource simulation software. This open-source software is authorized by Python scientifiic distributions. The Python software package includes numpy,openCV,PyAutoGUI library modules. These modules are license free and available as open tools. Each module is integrated in Python kernel, and Python programming language is used to simulate the proposed work.The purpose of this project is to recognize hand gesture with more accuracy. The design is very simple and the user doesn't need to wear any type of hand gloves Although this recognition application can be run in an ordinary computer having a web camera.These results part is focused on simulation part.

We got below results after implementing ,

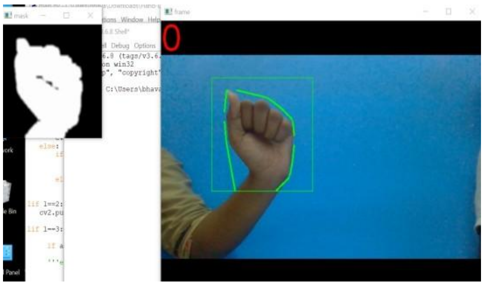
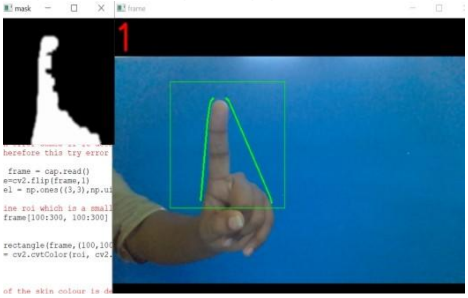


Figure 6.1 Recognizing gesture showing 0



```
herefore this try error

 frame = cap.read()
e=cv2.flip(frame,1)
el = np.ones((3,3),np.ui

ine roi which is a small
frame[100:300, 100:300]


rectangle(frame,(100,100
= cv2.cvtColor(roi, cv2.


of the skin colour is de
```

Figure 6.2 Recognizing gesture showing 1

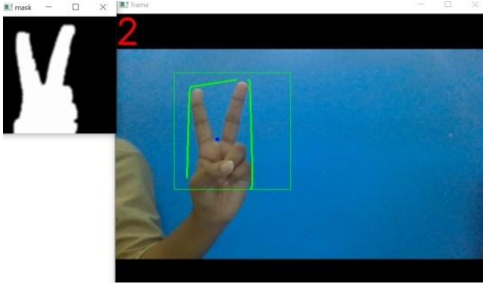


Figure 6.3 Recognizing gesture showing 2

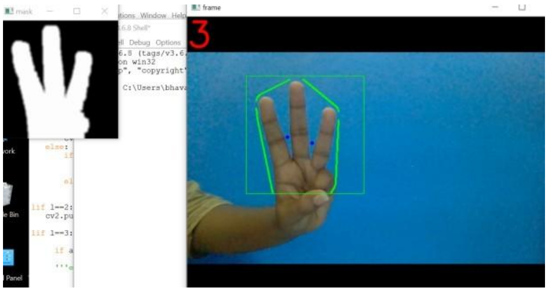


Figure 6.4 Recognizing gesture showing 3

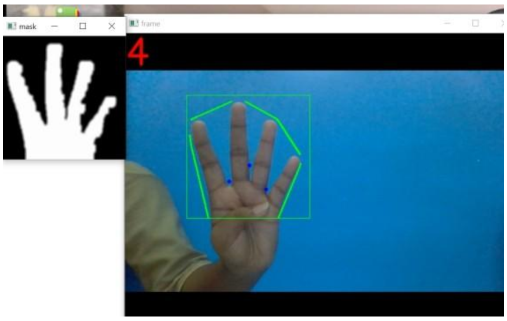


Figure 6.5 Recognizing gesture showing 4

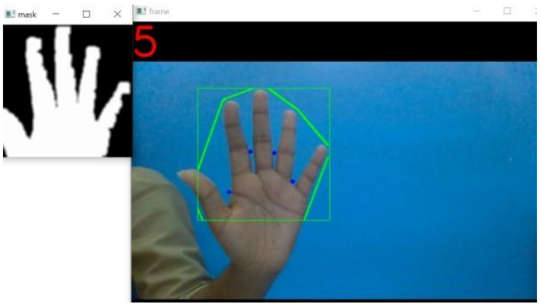


Figure 6.6 Recognizing gesture showing 5

In this we identified 0-5 numbers in this project we also identified some gesture like OK.Figures 6.1 -6.6 shows the zero to five fingers recognition.Below Figure 6.7 shows that OK symbol.

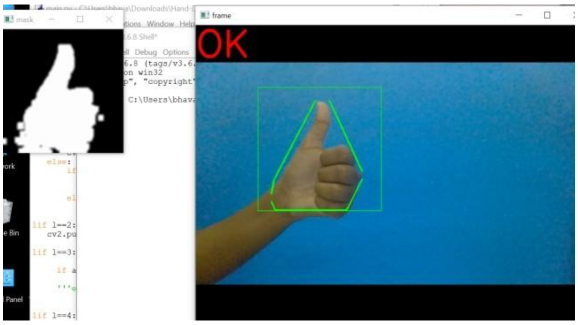


Figure 6.7 Recognizing gesture showing 1

In this method we got 98.7% accuracy which quite higher than the previous method edge detection, which has accuracy 78.81%.Recognition rate of proposed method is 96.6% where previous method has recognition rate is 85.7%.

Accuracy=number of predicted gestures/total number of gestures

Recognition rate=number of identified gestures/total number of gestures.

Method	Accuracy	Recognitionrate
Edge Detection	78.1	85.7
Background subtraction	98.7	96.6

Table 1: Comparision of accuracy and recognition rate

VII. CONCLUSION

As a conclusion based on the result of the project, it can be seen that developing the hand gesture recognition using Python and OpenCV can be implemented by applying the theories of hand segmentation and the hand detection system.To summarize it, this system has accomplished several objective in this project: (1) manage to establish a complete system for detecting, recognizing and interpreting hand gesture recognition through computer vision using Python and OpenCV, and (2) able to create the numbers and sign languages of hand gesture shown in the system that will meets the name of the project.

For the future recommendation, this system will include the execution of additional gestures that will allow any users with different skin colour and size of palm to perform more functions easily. The current system only uses the right hand with specific area in ROI to perform gestures. Therefore, the desired enhancement of the technique may possibly using both of user hands to perform different signs with computer operations. Additionally, background subtraction algorithms can be used for more effective performance.

REFERENCES

[1]. P.S.Neethu,R.Suguna,Divya Sathish,"An efficient method for Human Hand Gesture Detection and Recognition using Deep Learning Convolutional Neural Networks" , Springer-Verlag GmbH Germany, part of Springer Nature 2020,March 2020.

[2]. .Ashfaq T, Khurshid K (2016) Classifification of hand gestures using gabor fifilter with bayesian and nai¨ve bayes classififier. Int J Adv Comput Sci Appl 7(3):276– 279

[3]. Chaikhumpha T, Chomphuwiset P (2018) Real—time two hand gesture recognition with condensation and hidden Markov models. In: 2018 international workshop on advanced image technology (IWAIT), Chiang Mai, 2018, pp 1–4

[4]. Elmezain M, Al-Hamadi A, Michaelis B (2010) A robust method for hand gesture segmentation and recognition using forward spotting scheme in conditional random fifields. In: Proceedings of the 20th international conference on pattern recognition (ICPR'10), August 2010, pp 3850–3853

[5]. Hong J, Kim ES, Lee H-J (2012) Rotation-invariant hand posture classifification with a convexity defect histogram. In: IEEE international symposium on circuits and systems (ISCAS) 20–23 May 2012, pp 774–777

[6]. Kapil Y, Bhattacharya J (2016) Real-time hand gesture detection and recognition for human computer interaction. In: Berretti S, Thampi S, Srivastava P (eds) Intelligent systems technologies and applications. Springer, Cham, pp 559–567

[7]. Kawulok M, Grzejszczak T, Nelapa J, Knyc M (2012) Database for hand gesture

recognition. http://sun.aei.polsl.pl/*mkawulok/ gestures/. Accessed 12 May 2018

[8]. Liu K, Chen C, Jafari R, Kehtarnavaz N (2014) Fusion of inertial and depth sensor data for robust hand gesture recognition. Sens J IEEE 14(6):1898–1903

[9]. S. M. M. Roomi, R. J. Priya, and H. Jayalakshmi 2010 *Hand Gesture Recognition for Human- Computer Interaction* (J. Comput. Science vol. 6) no. 9 pp. 1002– 1007.

[10]. S. N. Karishma and V. Lathasree 2014 *Fusion of Skin Color Detection and Background Subtraction for Hand Gesture Segmentation* (International Journal of Engineering Research and Technology) vol. 3 no 1 pp 13–18.

[11]. A. Dhawan and V. Honrao 2013 *Implementation of Hand Detection based Techniques for Human Computer Interaction* (International Journal of Computer Applications) vol. 72 no. 17 pp 6– 13

[12]. C. Von Hardenberg and F. Bérard 2001 *Bare-hand human-computer interaction* (Proceedings of the 2001 workshop on Perceptive user interfaces) pp 1–8.

[13]. K. Nickel and R. Stiefelhagen 2007 *Visual recognition of pointing gestures for human–robot interaction* (Image Vis. Comput.) vol. 25 no. 12 pp 1875–1884

[14]. R. Lockton 2002 *Hand gesture recognition using computer vision* (4th Year Proj. Rep.) pp 1–69

[15]. Itseez 2017 *Open Source Computer Vision Library i tseez2015opencv* (Online) Available: https://github.com/itseez/opencv (Accessed: 13-Oct-2017)

[16]. Itseez 2017 *The OpenCV Reference Manual itseez2014theopencv 2.4.9.0* (Online) Available: http://opencv.org (Accessed: 13-Oct-2017)

[17]. S. van der Walt, S. C. Colbert, and G. Varoquaux 2011 *The NumPy array: a structure for efficient numerical computation* (Comput. Sci. Eng.) vol. 13 no. 2 pp 22– 30

[18]. Travis E. Oliphant et al 2017 *NumPy Developers numpy 1.13.0rc2*. (Online) Available: https://pypi.python.org/pypi/numpy (Accessed: 13-Oct-2017).

[19]. Just, A.: Two-handed gestures for human–computer interaction. IDIAPRR 6(73), 94 (2006)

[20]. Symeonidis, K.: Hand gesture recognition using neural networks. Neural Netw. 13(5), 1 (1996)

[21]. Pavlovic, V.I., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human–computer interaction: a review. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 677–695 (1997). doi:10. 1109/34.598226

[22]. Wu, Y., Huang, T.S.: Vision-based gesture recognition a review. In: Braffort, A., Gherbi, R., Gibet, S., Teil, D., Richardson, J. (eds.) Gesture-Based Communication in Human–Computer Interaction, vol. 1739, pp. 103–115. Springer, Berlin (1999)

[23]. Dadgostar, F., Barczak, A.L., Sarrafzadeh, A.: A color hand gesture database for evaluating and improving algorithms on hand gesture and posture recognition. Res. Lett. Inf. Math. Sci. 7, 127– 134 (2005).

[24]. Zhigang, F. Computer gesture input and its application in human computer interaction. Mini Micro Syst. 19996, 418–421.

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

[26]. Ahuja, M.K.; Singh, A. Static vision based Hand Gesture recognition using principal component analysis. In Proceedings of the 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE), Amritsar, India, 1– 2 October 2015; pp. 402–406.

[27]. Kramer, R.K.; Majidi, C.; Sahai, R.; Wood, R.J. Soft curvature sensors for joint angle proprioception. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 1919–1926.

[28]. Jesperson, E.; Neuman, M.R. A thin film strain gauge angular displacement sensor for measuring finger joint angles. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, New Orleans, LA, USA, 4–7 November 1988; pp. 807–vol. 6. Fujiwara, E.; dos Santos, M.F.M.; Suzuki, C.K. Flexible optical fiber bending transducer for application in glove-based sensors. IEEE Sens. J. 2014, 14,3631–3636. [CrossRef]

[29]. Shrote, S.B.; Deshpande, M.; Deshmukh, P.; Mathapati, S. Assistive Translator for Deaf & Dumb People. Int. J. Electron. Commun. Comput. Eng. 2014, 5, 86–89.

[30]. Gupta, H.P.; Chudgar, H.S.; Mukherjee, S.; Dutta, T.; Sharma, K. A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. IEEE Sens. J. 2016, 16, 6425–6432. [CrossRef]

[31]. Lamberti, L.; Camastra, F. Real-time hand

gesture recognition using a color glove. In Proceedings of the International Conference on Image Analysis and Processing, Ravenna, Italy, 14–16 September 2011; pp.365–373.

[32]. Wachs, J.P.; Kölsch, M.; Stern, H.; Edan, Y. Vision-based hand-gesture applications. Commun. ACM 2011, 54, 60–71. [CrossRef]

[33]. Pansare, J.R.; Gawande, S.H.; Ingle, M. Real-time static hand gesture recognition for American Sign Language (ASL) in complex background. JSIP 2012, 3, 22132. [CrossRef

[34]. Fang, Y.; Wang, K.; Cheng, J.; Lu, H. A Real-Time Hand Gesture Recognition Method. In Proceedings of the Multimedia and Expo, IEEE International Conference On Multimedia and Expo, Beijing, China, 2–5 July 2007. [CrossRef]

[35]. Oudah, M.; Al-Naji, A.; Chahl, J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. J. Imaging 2020, 6, 73. [CrossRef]

[36]. Al-Hammadi, M.; Muhammad, G.; Abdul, W.; Alsulaiman, M.; Bencherif, M.A.; Alrayes, T.S.; Mekhtiche, M.A. Deep learningbased approach for sign language gesture recognition with efficient hand gesture representation. IEEE Access 2020, 8, 192527–192542. [CrossRef]

[37]. Vaitkeviˇcius, A.; Taroza, M.; Blažauskas, T.; Damaševiˇcius, R.; Maskeliunas, R.; Wo´zniak, M. Recognition of american sign ¯ language gestures in a virtual reality using leap motion. Appl. Sci. 2019, 9, 445. [CrossRef]

[38]. Rezende, T.M.; Almeida, S.G.M.; Guimarães, F.G. Development and validation of a brazilian sign language database for human gesture recognition. Neural Comput. Appl. 2021. [CrossRef]

[39]. Žemgulys, J.; Raudonis, V.; Maskeliunas, R.; Damaševiˇcius, R. Recognition of basketball referee signals from real-time videos. ¯ J. Ambient Intell. Humaniz. Comput. 2020, 11, 979–991. [CrossRef]

[40]. Afza, F.; Khan, M.A.; Sharif, M.; Kadry, S.; Manogaran, G.; Saba, T.; Ashraf, I.; Damaševiˇcius, R A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection. Image Vision Comput. 2021, 106, 104090. [CrossRef]