Convexity Defect Based Hand Gesture Recognition Using OpenCV

Md. Najmus Salehin*, Md. Helal An Nahiyan, Fahim Islam Anik
Department of Mechanical Engineering, Khulna University of Engineering & Technology, Khulna-9203, Bangladesh

Abstract. The technology of gesture recognition systems focuses on improving human-machine interaction and proximity control. This paper mainly eyes on the hand gesture recognition system that can be used for controlling different automated systems such as automated wheelchairs for physically impaired people, robotic manipulators, and similar mechanisms. Out of several techniques, a convexity defect-based system is discussed here. As the platform, Spyder is used, which is an open-source cross-platform integrated development environment (IDE). The OpenCV library is used, which is a library of functions focusing on real-time computer vision. After developing the system for multiple features of different finger-based gestures, the distance-based accuracy for each feature is measured, where at the most effective distance (2ft) the average accuracy is found to be 93.75% with real-time response. In the case of the following technique, the gesture orientation is not mattered so the system is kept simple and time-efficient.

Keywords: Hand gesture recognition, convexity hull defect, human-machine interaction, Python, Spyder, OpenCV

Introduction

In this fast-growing world of technology, the efficacy of console-based control systems is being questioned more than before. The large array of gesture recognition and other sensory methods are taking place of conventional methods when it comes to the controlling of various systems. It has a large array of applications including virtual environment control, sign language translation, robot remote control, or musical creation (Allevard, Benoit, & Foulloy, 2006). There are various techniques of gesture recognition. Non-vision and vision-based approaches are used to achieve hand gesture recognition (Haria *et al.*, 2017). Among the non-vision techniques, the wired fuzzy gloves technique is quite popular where different sensors like accelerometers and gesture sensors are used to detect motion. Vision-based techniques include static (Bagdanov *et al.*, 2012) and dynamic (De Smedt, Wannous & Vandeborre, 2016; Data, 2013) gestures.

The static gesture recognition technique is suitable where a stationary signal is given to the machine, on the contrary, the dynamic technique involves motion-based gestures. In the newest dimension of human-machine interaction (HMI), gesture recognition systems surely are transcending the interaction method from a machine-centered system to a human-centered system.

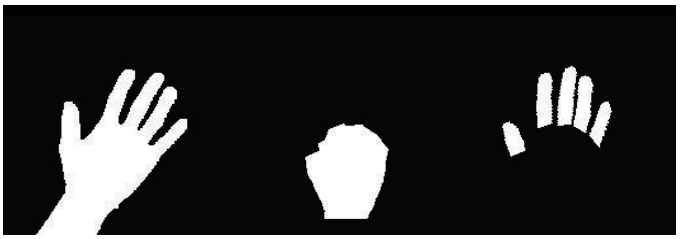
Among the static vision-based techniques, the convexity hull defect method is one of the most popular ones. Others being brightness factor matching (Hasan & Mishra, 2010), neural networking (Strezoski *et al.*, 2019), fingertip and palm area detection (Marin *et al.*, 2013), and so on. Convexity hull defect is theoretically based on cosine theorem and defects are found based on the angles between the fingers.

This paper provides the methodology of a gesture recognition system based on the convexity defect method with four gesture features. These features can be used to control a wheelchair for starting, stopping, turning right, and turning left. Some test results are evaluated to assess the accuracy of gesture recognition.

* Corresponding Author

Related Works

Several works have been done on gesture recognition for both vision and non-vision cases using various techniques. Chen *et al.* (2014) proposed a very efficient technique of segmenting palm and fingers to recognize gestures made by the fingers. They applied a rule classifier to predict the labels of the gesture. After segmenting the palm area, the labeling algorithm marks the finger region (shown in Figure 1).



(a) (b) (c)

Figure 1. a) The detected hand region, b) Palm mask c) Segmented fingers (Chen *et al.*, 2014)

Hasan and Mishra (2010) proposed a system for gesture recognition based on the block local brightness of gesture image. Strezoski *et al.* (2019) proposed a convolutional neural network-based system by training a dataset for a robust deep model.

For dynamic systems, Kurakin, Zhang and Liu (2012) developed a system for dynamic gesture recognition using depth sensor and action graph-based training. De Smedt, Wannous and Vandeborre (2016) proposed a skeleton-based dynamic gesture recognition system for 3D gestures using hand geometry (shown in Figure 2). Wang *et al.* (2012) used Hidden Markov Models (HMMs) (Rabiner, 1989) to develop a dynamic gesture recognition system.

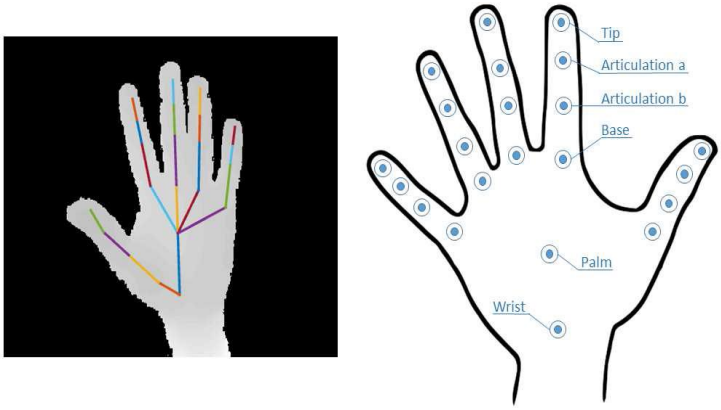


Figure 2. Depth and hand skeleton dataset (De Smedt, Wannous & Vandeborre, 2016)

Some of the works also focus on non-vision-based gesture recognition. Luzhnica *et al.* (2016) worked on a data glove to recognize hand signals. The custom data glove is shown in Figure 3. That data glove was equipped with different sensors. Luzanin and Plancak (2014) proposed a low-budget data glove for gesture recognition system trained by a cluster-based probabilistic neural network (PNN).

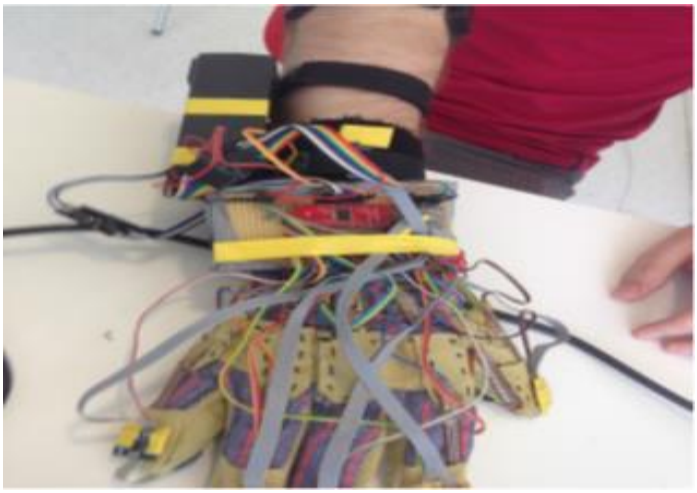


Figure 3. Custom data glove (Luzhnica *et al.*, 2016)

Focusing on the previous studies, this research intended to develop a simplified gesture recognition system using the OpenCV platform to control different automated mechanisms especially the wheelchair developed for physically impaired people (Ahmed, Karim & Nahiyan, 2015). To achieve the intended goal, a technique called convexity defect recognition is considered, which provides the feature recognition information based on the defects in the convex hull.

Methodology

A hand gesture recognition system is developed based on the Convexity defect method. Figure 4 shows the system flowchart:

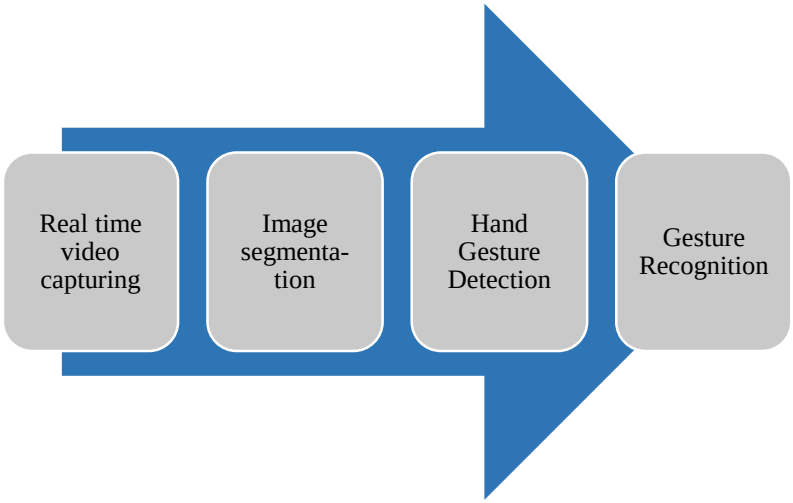


Figure 4. System flowchart

Capturing real-time video

At first, the real-time video is captured. For that, a range of pixels is chosen to define the area of interest. A 100x300 pixels sized resolution is set to define the area of interest. The captured video feeds back images in RGB scale which is converted in greyscale in the hand image segmentation section. The area of interest is confined in a rectangle.

Hand image segmentation

At this stage, the hand feature is obtained from the real-time video. A filtering process for skin detection is applied. At first, the video feature is filtered to be able to detect hand

features. To detect a feature in a specific skin color range, the real-time video feature is first converted from RGB (Red, Green, and Blue) pixels to grey (Image Processing Basics — RGB-to-grayscale projection). "CV_RGB2GRAY" conversion function is used to perform that. The conversion is executed by the function method using the following equation:

$$.30R + .59G + .11B = Val \quad (1)$$

The next step is thresholding (Henden, 2004). Where the greyscale image is converted to a monochrome image to execute detection. Otsu's binarization method (Otsu, 1996) is used for thresholding in the current work. This algorithm classifies the pixels into two classes by returning a single intensity threshold value, foreground, and background. This threshold is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance (Otsu, 1996). The following equation is followed by the threshold operation.

The weighted class variance:

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t) \quad (2)$$

The estimated class probability:

$$q_1(t) = \sum_{i=1}^{t} P(i) \;\&\; q_2(t) = \sum_{i=t+1}^{l} P(i) \quad (3)$$

Class means:

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{w_1(t)} \;\&\; \mu_2(t) = \sum_{i=t+1}^{l} \frac{iP(i)}{w_2(t)} \quad (4)$$

After performing Gaussian blur for the greyscale feature, Otsu's binarization method is executed by the function "cv2.threshold". This function takes 4 arguments- the blurred image, threshold value of the pixels (127), the maximum value of the pixels (255), and conversion command "cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU".

Another type of image segmentation strategy is followed by only masking without any thresholding as an alternative, where the lower and the upper pixel value is set and anything within this lower and upper range is considered as skin color and the pixel value is set to 255 and anything out is considered not as skin and the pixel value is set to 0. Figure 5 shows the two types of segmented images.

(a) (b)

Figure 5. (a) Thresholded image (b) masked image (Current Paper)

Detection

Contours:

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having the same color or intensity. The contours are a useful tool for shape analysis, object detection, and recognition (OpenCV, 2016). By using the "cv2.findContours" function, the object is detected by creating an array of continuous points along the boundary of an object, as all the pixels covering the boundary of a single object likely to have the same color and

intensity range. A contour hierarchy (OpenCV, 2015) is set in case multiple coexisting objects are subjected. "RETR_TREE" flag is used as contour retrieval mode.

Convex hull

Any region or shape to be convex the line joining any two points belonging to the region is to be contained entirely in that region. For a given shape or set of points, many convex curves can be found for each boundary. The smallest or the tight-fitting convex boundary is known as a convex hull (Kang & Atul, n.d.). Figure 6 shows the main concept behind the convex hull. For application in OpenCV library, it provides a built-in function called "cv2.convexHull". It takes 3 arguments: points (the main feature that it is to be operated, in this case, the contour of the hand), clockwise (determines the output hull is clockwise or counterclockwise oriented), and return points (returns the coordinate of the points). Figure 9 shows the contour and convex hull developed in the current thesis.

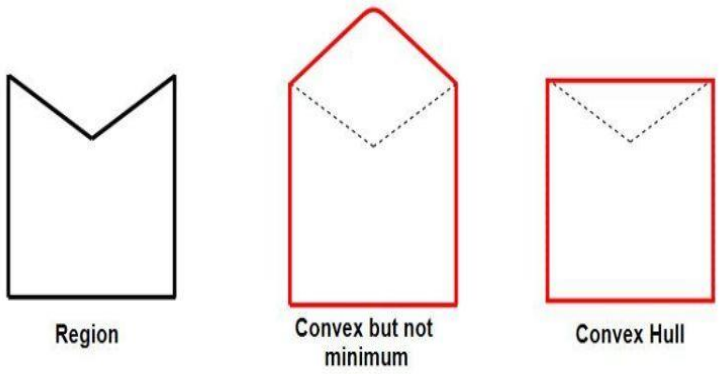


Figure 6. Convex hull idea (Kang & Atul)

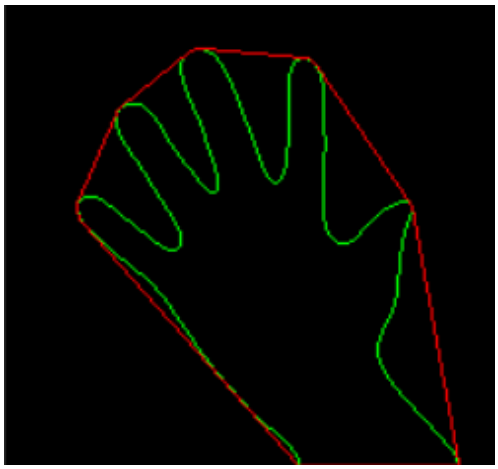


Figure 7. Contour (green) and convex hull (red) (Current thesis)

Convexity defect:

Convexity defect is the defect or deviation caused between contour and convex hull. Any deviation of the object from its hull can be considered a convexity defect (OpenCV, 2013). Figure 8 illustrates the concept of a convexity defect using an image of a human hand (Bradski & Kaehler, 2009).

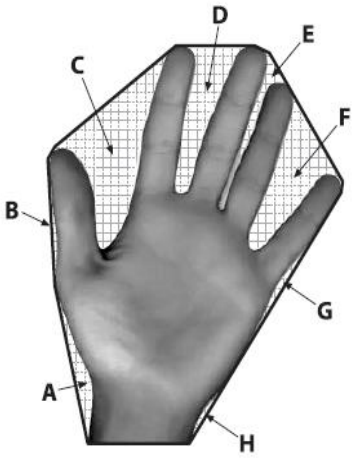


Figure 8. Convexity defects (Bradski & Kaehler, 2009)

Here the outermost feature represented by the dark line is the convex hull, the gridded areas (A-H) represent defects in the hand contour relative to the convex hull. Figure 9 shows the idea of detecting the defect. Where the triangle produced by starting point "C", endpoint "B", and the farthest point "A" is assessed for the angle associated with the farthest point. The formula of the cosine theorem is used.

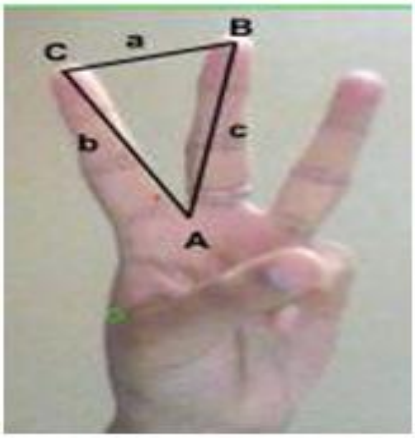


Figure 9. Convexity defect detection (Marium *et al.*, 2017)

The length of each line is found by the distance formula. Then cosine formula is used to determine the angle.

$$a^2 = b^2 + c^2 - 2bc\cos A \quad (5)$$

The angle is:

$$A = cos^{-1}\left(\frac{b^2+c^2-a^2}{2bc}\right) \quad (6)$$

Any angle under 90 degrees is set to be a defect which is executed by the function "cv2.convexHull". Figure 10 shows the convexity defect detection for current work by a red dot between two fingers.

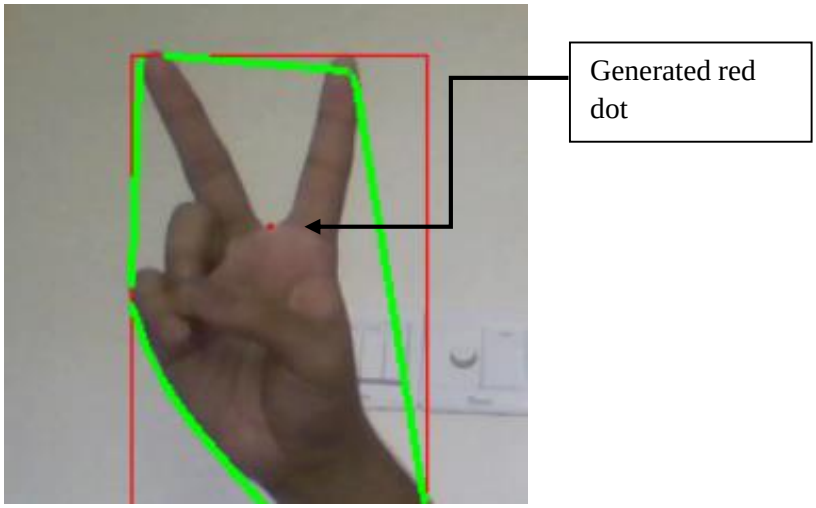


Figure 10. Defect detection

Gesture recognition:

After the hand contour, convex hull hence the convexity defect is detected, the gesture recognition part is pretty much straightforward. Based on the number of defects various gesture signals are assigned. In the current thesis, 4 different gestures are worked with. Figures 11-14 show the proposed features in this paper.

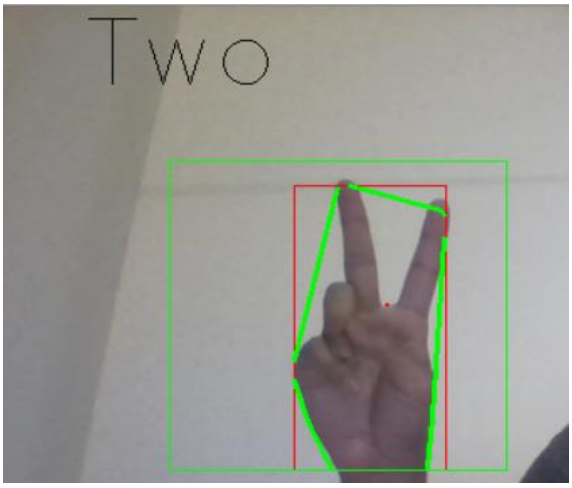


Figure 11. Gesture feedback "Two" for 1 defect

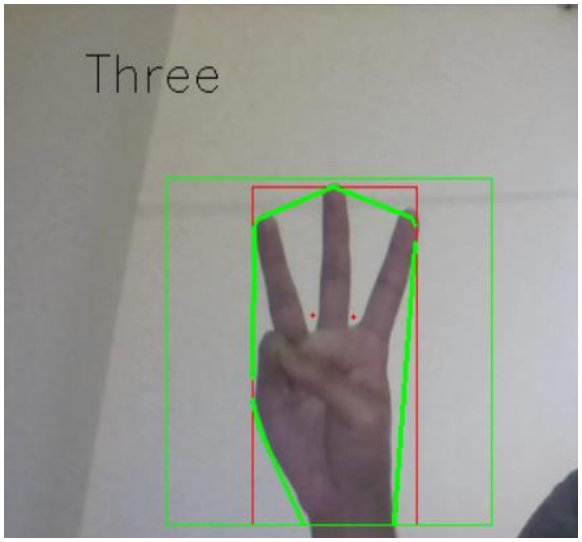


Figure 12. Gesture feedback "Three" for 2 defects

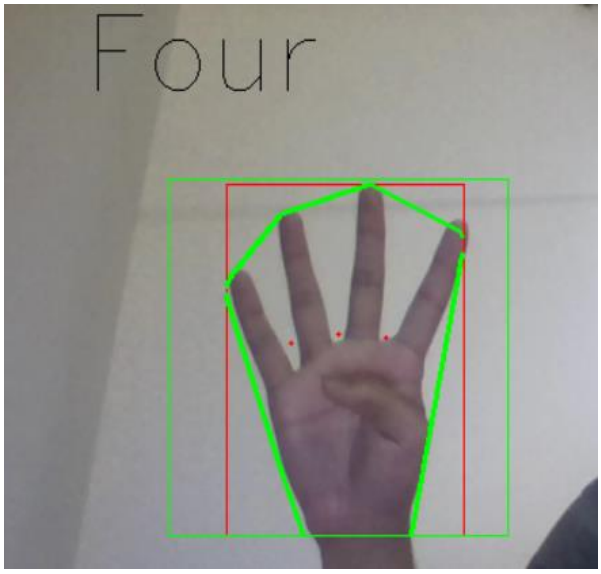


Figure 13. Gesture feedback "Four" for 3 defects

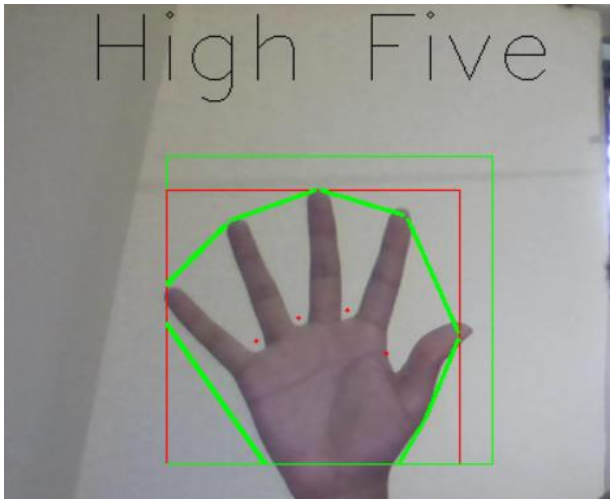


Figure 14. Gesture feedback "High Five" for 4 defects

Results and Discussion

In the current paper, 4 different gesture features are proposed. For the accuracy test, various approaches are taken into account. Firstly, the distance-based accuracy in the case of every gesture feature is tested with plain background from 1ft to 3ft. Tests are performed at every 0.5ft distance. The result shows that the best accuracy is achieved at the 2ft distance with an average 93.75% accuracy, as 100 observations are made for each case.

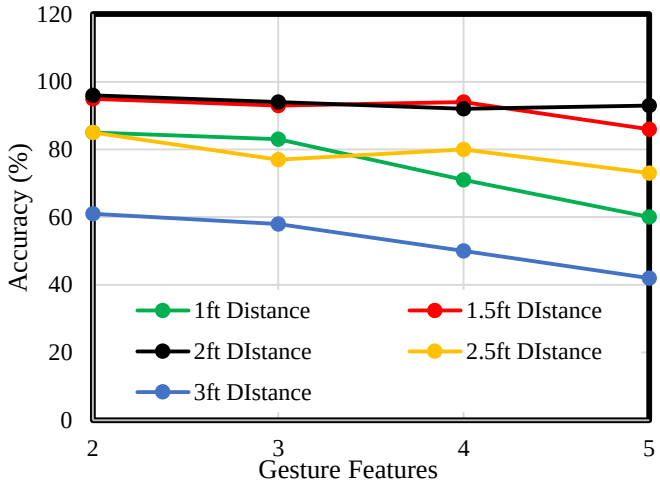


Figure 15. Variation of accuracy for each features with distance

In Figure 15, by 2, 3, 4, 5, the proposed features are defined, respectively, "Two", "Three", "Four" & "high Five". Figure 16 shows the average accuracy of the features for each distance.

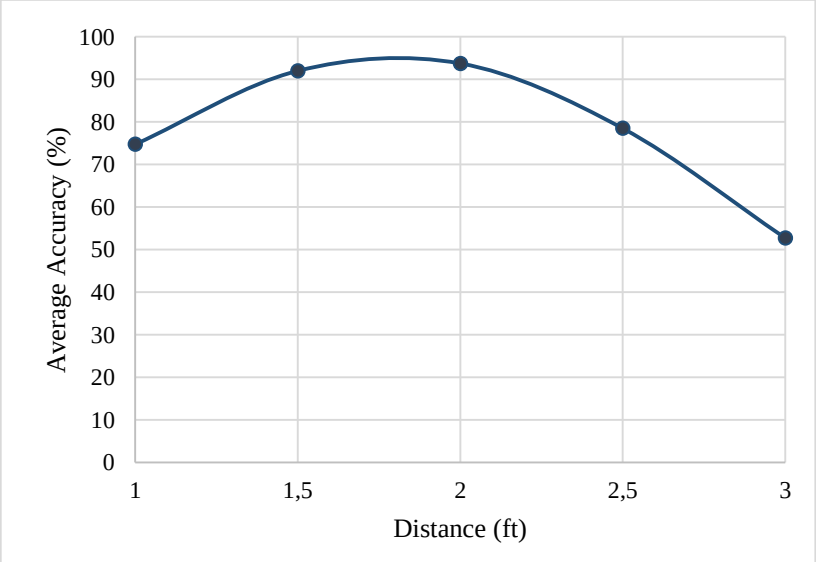


Figure 16. Average accuracy for each test distance

The comparison between the threshold system and the masked system is shown in Figure 17 (for distance = 2ft). It shows that a thresholded system is far superior to a system where only masking is used.

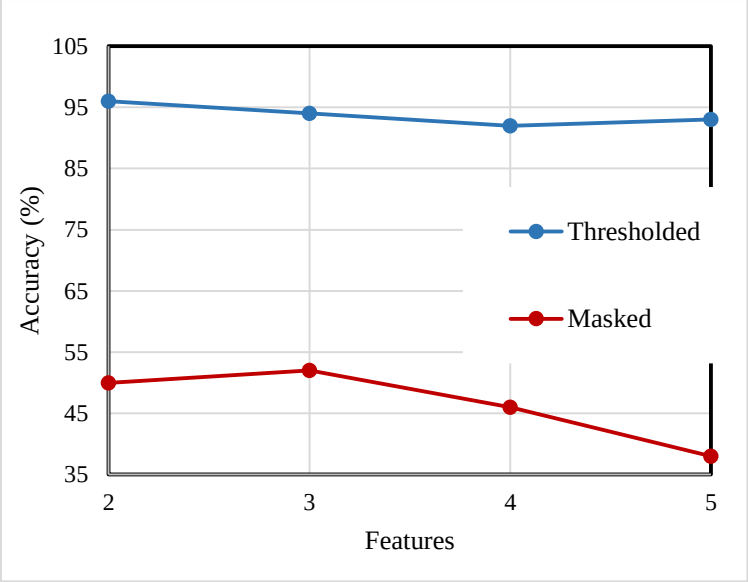


Figure 17. Thresholded vs. masked system

Another comparison is made for correspondent features with the work by Xu, Park and Pok (2017). Where Euclidean distance matric is used and hand centroid is found. The defect angle is measured from the centroid. For the comparison, the most effective distances (1.5ft and 2ft) are used to average the accuracy.

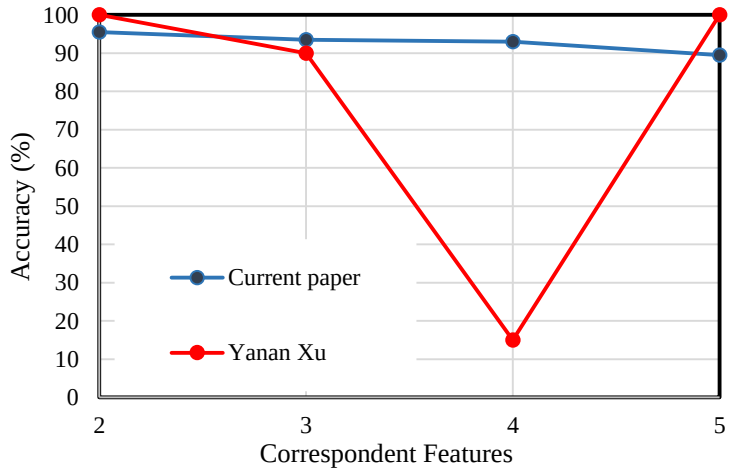


Figure 18. Current Paper vs. Y. Xu

Figure 18 shows that Yanan Xu developed a system that is effective for most of the features, but for some features, it loses effectiveness (for example, feature 4). On the contrary, the current system maintains a uniform accuracy for the features.

The main challenging part of this work is background subtraction because contour line is developed by joining the points for each pixel having the same intensity, backgrounds with high-intensity color and high-intensity light tempers the fact of having required and controlled contour of the hand, so, accuracy falls drastically with random backgrounds.

Table 1. Accuracy with plain and random background (2ft Distance)

Gesture Features	Convexity defect	%Accuracy (Plain BG)	% Accuracy (Random BG)
"Two"	1	96	43
"Three"	2	94	46
"Four"	3	92	40
"High Five"	4	93	38

Table 1 suggests a huge fall in the performance when the system is tested before random backgrounds. This performance can be improved with more processed hand image segmentation and higher camera quality.

For detecting convexity defect, the angle is set at 90 degrees, so, for any angle over 90 degrees, no convexity defect is accounted. It is observed that the system does not recognize defects under 20 degrees (averaged for different distances). Figure 19 shows the angular range for convexity defect detection, where 1 represents defect and 0 represents no defect.

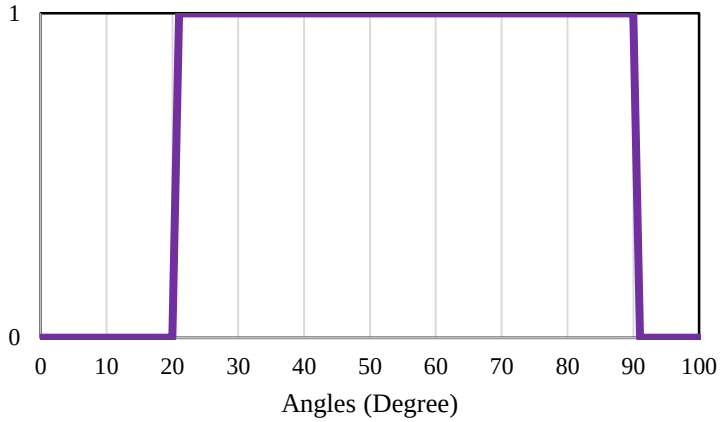


Figure 19. Angular Range for defect detection

Conclusion

In this new era, when technology dominates everywhere innovation is the key to success. To execute various distant operations, gesture-based systems are effective ones, especially where the operations are harmful to humans to operate manually. In this work, a vision-based static gesture recognition system is developed to control a robotic system such as a wheelchair for physically challenged people. The findings in the performance and accuracy test were satisfying enough to understand its validity in real-life application. It is noted that a thresholded system is much more effective than the only masked system. The work done by Xu, Park and Pok (2017) is effective as a many variation feature system but it may be proved to be less effective for some features as only depth is used to classify the features. Therefore, the developed system is valid enough for real-life applications, whether the application is control-based, security-enhancing and detecting or any other field-based.

Nomenclature

σ: class variances
w: weights according to q
q: class probability
μ: class means
t: threshold

References

Ahmed, H., Karim, K. E. & Nahiyan, H. (2015). Design, Simulation and Construction of an Automatic Wheelchair. *International Conference on Mechanical, Industrial and Materials Engineering 2015* (ICMIME2015), Paper ID: AM 19, 11-13 December, 2015, RUET, Rajshahi, Bangladesh. doi: 10.13140/RG.2.1.1606.5045.

Allevard, T., Benoit, E. & Foulloy, L. (2006). Hand posture recognition with the fuzzy glove. In B. Bouchon-Meunier, G. Coletti, & R. R. Yager (Eds.), *Modern Information Processing* (pp. 417–427). Elsevier Science. doi: 10.1016/B978-044452075-3/50035-2.

Bagdanov, A. D., Del Bimbo, A., Seidenari, L., & Usai, L. (2012). Real-time hand status recognition from RGB-D imagery. *Proceedings of the 21st International Conference on Pattern Recognition* (ICPR2012), pp. 2456-2459. IEEE.

Bradski, G. & Kaehler, A. (2009). Learning OpenCV—Computer Vision with the OpenCV Library. *IEEE Robotics and Automation Magazine, 16*(3). doi: 10.1109/MRA.2009.933612.

Chen, Z. H., Kim, J. T., Liang, J., Zhang, J., & Yuan, Y. B. (2014). Real-time hand gesture recognition using finger segmentation. *The Scientific World Journal, 2014*, 820–824. doi: 10.1155/2014/267872.

Data, R. U. S. A. (2013). (12) Patent Application Publication (10) Pub. No.: US 2013 / 0344194A1.

De Smedt, Q., Wannous, H. & Vandeborre, J. P. (2016). Skeleton-Based Dynamic Hand Gesture Recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1206–1214. doi: 10.1109/CVPRW.2016.153.

Haria, A. *et al.* (2017). Hand Gesture Recognition for Human Computer Interaction. *Procedia Computer Science*, *115*, 367–374. Elsevier B.V. doi: 10.1016/j.procs.2017.09.092.

Hasan, M. & Mishra, P. K. (2010). HSV Brightness Factor Matching for Gesture Recognition System. *International Journal of Image Processing (IJIP)*, *4*(5), 456–467.

Henden, C. (2004). Exercise in Computer Vision. A Comparison of Thresholding Methods. *Ntnu*, (November), 1–7.

Image Processing Basics — RGB-to-grayscale projection (no date). Retrieved September 24, 2020, from http://www.imageprocessingbasics.com/rgb-to-grayscale/

Kang & Atul (n.d.). *Convex Hull — OpenCV 2.4.11.0 documentation*. Retrieved September 24, 2020, from https://theailearner.com/tag/convex-hull-opencv/

Kurakin, A., Zhang, Z., & Liu, Z. (2012). A real time system for dynamic hand gesture recognition with a depth sensor. Moscow Institute of Physics and Technology, Department of Applied Math and Control, Moscow, (Eusipco), pp. 1975–1979.

Luzanin, O. & Plancak, M. (2014). Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network. *Assembly Automation*, *34*(1), 94–105. doi: 10.1108/AA-03-2013-020.

Luzhnica, G. *et al.* (2016). A sliding window approach to natural hand gesture recognition using a custom data glove. *2016 IEEE Symposium on 3D User Interfaces, 3DUI 2016 - Proceedings*, (March), pp. 81–90. doi: 10.1109/3DUI.2016.7460035.

Marin, G., Fraccaro, M., Donadeo, M., Dominio, F., & Zanuttigh, P. (2013, October). Palm area detection for reliable hand gesture recognition. *Proceedings of MMSP, 2013*, p. 120. Retrieved from: http://lttm.dei.unipd.it/nuovo/Papers/13_MMSP_palm_detection.pdf.

Marium, A. *et al.* (2017). Hand Gesture Recognition Using Crf. *International Journal of Advance Engineering and Research Development*, *4*(04), 90–94. doi: 10.21090/ijaerd.co011.

OpenCV (2013). *Contours: More Functions*, *OpenCV-Python Tutorials*. Retrieved September 24, 2020, from https://docs.opencv.org/3.4/d8/d1c/tutorial_js_contours_more_functions.html

OpenCV (2015). *OpenCV: Contours Hierarchy*. Retrieved September 24, 2020, from https://docs.opencv.org/master/d9/d8b/tutorial_py_contours_hierarchy.html

OpenCV (2016). *OpenCV: Contours: Getting Started*, *OpenCV*. Retrieved September 24, 2020, from https://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html

Otsu N. (1996). A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man and Cybernetics*, *9*(1), 62–66.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE, 77*(2), 257-286. doi: 10.25300/MISQ/2017/41.3.08.

Strezoski, G., Stojanovski, D., Dimitrovski, I., & Madjarov, G. (2019). Hand Gesture Recognition Using Convolutional Neural Networks. *2018 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium)*, USNC-URSI 2018 - Proceedings, (September), pp. 147–148. doi: 10.1109/USNC-URSI.2018.8602809.

Wang, X. *et al.* (2012). Hidden-Markov-Models-based dynamic hand gesture recognition. *Mathematical Problems in Engineering*, *2012*. doi: 10.1155/2012/986134.

Xu, Y., Park, D. & Pok, G. (2017). Hand Gesture Recognition Based on Convex Defect Detection. *International Journal of Applied Engineering Research*, *12*(18), 7075–7079. doi: 10.1109/ISACV.2018.8354074.