HAND GESTURE RECOGNITION METHOD

Table of Contents

ABSTRACT:

Gesture recognition is "the entire procedure of tracking gesture to their representation and converting them to some purposeful command." Gesture recognition turns up to be important field in the recent years. Communication through gestures has been used for many other applications.

This hand gesture method is used for various purposes like human computer interactions, robotics, sign language recognition, controlling a music player, playing game, air browsing etc. Hand gestures replace peripheral devices like keyboard, mouse with hand gestures for human-computer interaction (HCI). Hand gesture recognition basically involves four steps – image acquisition, hand segmentation, feature extraction and classification. Mostly hand gesture techniques have been divided into two categories- sensor based and vision-based techniques. In short, the hand gestures are captured through a camera in the devices. The captured image of hand gesture is then processed to perform the specific control functions. The processing is done by the pre-written program code. In this, each gesture sign is assumed for a function.

KEY WORDS: Image Acquisition, Thresholding, Contours, Convex hull, vision-based.

1. INTRODUCTION:

Gestures are the movement of any body part used to convey the meaningful information. Communication through gestures has been widely used by humans to express their thoughts and feelings. Human gesture is a mode of non - verbal interaction. Gestures recognition is the process of identifying gestures performed by human. Gestures have been classified in two categories static and dynamic. Static gestures refer to still body posture and dynamic refers to movement of body part. Gestures can be performed with any body part like head, face, arms, hands, etc.

The hand gestures are mainly used for the purpose of controlling the devices from a distance without the help of any intermediates like the remote. This is a direct interaction between the devices and the users. In general, the framework for hand gesture recognition includes taking in the input i.e. hand image, hand detection, preprocessing i.e. removing unwanted features, segmentation, feature extraction, recognition (gesture dictionary) and execution (commands).

Hand Gestures Recognition techniques have been divided into two categories- Sensor based and Vision Based recognition. Sensor based recognition collects the gesture data by using one or more different types of sensors. These sensors are attached to hand which record to get the position of the hand and then collected data is analyzed for gesture recognition. Eg data glove, Wii controller, EMG sensors, accelerometer sensors, etc. Sensor based recognition has certain limitations

a) It requires a proper hardware setup which is very expensive.
b) It hinders the natural movement of the hand.

So to overcome the limitation of sensor based recognition vision based techniques came into existence. Vision based techniques make use of camera to capture the image for hand gesture. Vision based recognition make use of many image processing algorithms to get hand posture information and movement of hand. This approach recognizes gesture from shapes, orientations, contours, and colour or motions features of a hand. Colour markers are an example of vision-based recognition. But the vision based recognition also has some limitations that it is affected by illumination changes and cluttered backgrounds.

Figure 1: Overview of Classification system

2. RELATED WORK:

Nishihara, et al [1] this paper includes a method of providing device inputs. It also illuminating hand gestures performed via a bare hand of a user in a foreground of a background Surface with at least one infra-red (IR) light source. Determining a plurality of three-dimensional features of the bare hand relative to the background surface based on a parallax separation of the bare hand in the first plurality of silhouette images relative to the second plurality of silhouette images. Providing at least one device input corresponding to interaction with displayed visual content based on the provided input gesture corresponding to one of the plurality of predefined gesture inputs.

Zhang, et al. [2] dynamic hand gestures are recognized by processing depth data, including in real-time. Phases involved in it are depth sensor, intended gesture, training data, hand

segmentation, data normalization (orientation / scale), feature extraction, gesture graph training, gesture graph, segmentation and tracking, normalization - orientation and scale. Segmentation technique: find body from blobs, find hand (hypotheses) from threshold, refine by removing arm if needed. No.of. Gestures used is 10. Features Extracted: velocity (from current - previous frame), rotation from saved parameters, shape descriptor(s) including dimensionality reduction.

Quentin De Smedt, et al[4] the performance of recent recognition approaches using a challenging hand gesture dataset containing 14 gestures, executing the same gesture with two different numbers of fingers. Two groups have taken part to this track, the accuracy of the recognition algorithms have been checked and compared to three other approaches. Two recognition methods have been registered to the track: i) Skeleton-based Dynamic hand gesture recognition [DSWV16] from IMT Lille Douai / University of Lille, France. ii) Classify sequence by key frames with convolutional neural network from ONERA and ENSTA ParisTech, France.

Aashni Hariaa, et al [5] in this paper a robust marker- less hand gesture recognition system which can efficiently track both static and dynamic hand gestures. System translates detected gesture into actions such as opening websites and launching applications like VLC Player and PowerPoint. It is used to shuffle through the slides in presentation. Image acquisition Method is done with webcam or USB based webcam. No. of Gestures used is 24 alphabets (upper and lower case) and 10 numbers. Recognition Method used is perpendicular (exact shape of Surface along with direction) or convexity (when determining which alphanumeric).

Yan et al [6] receiving input from a user includes providing a surface within reach of a hand of the user. Pluralities of locations on the surface that are touched by the user are sensed. Modules involved in this method are gesture capturing module, gesture recognition module, gesture confirmation module, communication module, navigation system, audio system, HVAC system. Features extracted are fingertips using curvature and convexity defects, Euclidean distance, and direction.

Gang Li, et al [7] it dynamic hand gesture recognition (radar sensor). The sparse representation of the radar signal in the time-frequency domain is achieved through the Gabor dictionary, and then the micro-Doppler features are extracted by using the orthogonal matching pursuit (OMP) algorithm and fed into classifiers for dynamic hand gesture

recognition. Method is validated with real data measured with K-band radar. Recognition Method: i) HMM for dynamic ii) SVM

Nowozin et al. [8] a gesture detection and recognition technique is described. In one example, a sequence of data items relating to the motion of a gesturing user is received. A selected set of data items from the sequence are tested against pre-learned threshold values, to determine a probability of the sequence representing a certain gesture. If the probability is greater than a predetermined value, then the gesture is detected, and an action taken. In examples, the tests are performed by a trained decision tree classifier. In another example, the sequence of data items can be compared to pre-learned templates, and the similarity between them determined. If the similarity for a template exceeds a threshold, a likelihood value associated with a future time for a gesture associated with that template is updated.

Arnon Amir, et al [9] they presented the first gesture recognition system implemented end-to-end on event-based hardware, using a TrueNorth neurosynaptic processor to recognize hand gestures in real-time at low power from events streamed live by a Dynamic Vision Sensor (DVS). The advantages of event-based sensors are diluted if their event streams must be cast back into synchronous frames for the benefit of conventional processors downstream. Conventional processors, like CPUs and GPUs, are efficient in processing dense, synchronously delivered data structures, not sparse, asynchronous event streams. Throughput is maintained by keeping the instruction and data pipelines as full as possible, even at the cost of executing redundant computations on unchanging data.

Eleni Tsironi, et al [10] gesture recognition is a field of particular interest to re- searchers varying from virtual reality to Human–Robot interaction. In this paper, we contribute to a more detailed description and analysis of the Convolutional Long Short-Term Memory Recurrent Neural Network (CNNLSTM) that we introduced in our former work to deal with gesture recognition. CNNLSTM was inspired by the efficiency of Convolutional Neural Networks in implicit feature extraction and the ability of Recurrent Neural Networks in modeling time series. We proposed CNNLSTM based on Multi- channel Convolutional Neural Networks (MCCNN) for gesture recognition and extended the concept by adding Long Short-Term Memory units for addressing the problem of dynamic gesture recognition as a sequence-labeling task and not as a simple classification task as in the case of MCCNN.

Vipul Mehra et al[11] This paper explores a very well-known technique for image classification and recognition that is Bag of Visual Words. The process involves feature extraction using Canny Edge Detector and Scale-Invariant Feature Transform (SIFT), codebooks construction using generative model like K-Means and Vector quantization. Finally, classification is done using Support Vector Machines (SVM) using chi-squared kernel. After applying 10 cross validation the accuracy comes to be around ≈ 70%.

physical gadget that goes about as input gadget for the framework that perceives motions.

Zhipeng Liu et al [13] in recent years, gesture recognition has gained a great deal of attention because of its great potential applications, such as sign language translation, human computer interactions, robotics, virtual reality and so on. The contribution of there work mainly lies in the following three aspects. Firstly, a novel two streams Faster RCNN (2S Faster R-CNN) hand detector is developed to get accurate hand regions by integrating RGB and depth inputs. Secondly, the hand-oriented C3D feature effectively characterizes gestures, including the hand postures and the motion trajectories.

Nirali A. Patel, et al [14] Movement of Human Hands interpreted mathematically by gesture recognition system. In our Modern life, gestural medium plays an important role for interaction. The major obstacle is the lack of knowledge about blind user's preferences toward hand gestures. Here, they aimed to use reduced shape signature[1] method for gesture recognition which is given by Gourav Modanwal, and Kishor Sarawadekar.Here I add some extra keyboard key mapping with intra class gesture[1] using classification rule and also add dynamic gesture.

Nirali A. Patel, [15] Gestures have for some time been considered as a collaboration strategy that can conceivably convey more innovative, pure and natural strategies for speaking with our computers. This paper gives an examination of provisional surveys done in this area. There are two noteworthy methods that are utilized for hand gesture Recognition, i) Vision based methods, ii) Contact based methods

Vision based methods are situated in light of association of the client with single or different camera setups. Contact Based methods depend on physical collaboration of the client with a

Zhuang, Huiwei, et al [16] Hand gesture recognition is more and more popular method. In this paper we are using the novel static gesture recognition method based on non-negative

matrix factorization (NMF) and compressing sensing (CS). .first we project the image to subspace using NMF then we use to design a classifier based on the CS classifier. The result indicates that the CS classifier can achieve a higher recognition rate and has a better occlusion resistance. The result also demonstrate that in resisting occlusion NMF perform better than PCA and so the recognition rate of the overall system is higher than any other method.

A.O. Alghabri, et al [17] in this paper is going to use the FPGA and AR method for hand gesture recognition. This method is mainly used for the purpose of pre-processor. In this paper the recognition and the tracking processes of hand gesture and marker based iterative multi-application AR system on FPGA to reduce the overall power consumption. In this we have used the 2D webcam to reduce the cost and the power consumption. This system is more efficient for children's by kid's practice, by using this system gets 93.2% of accuracy.

E I Nikolaev et al [18] in this paper is going to use the virtual data training using the deep model technique. This hand gesture paper has used the deeply trained model through the hand image. In this we have used the two deep convolution neutral networks. First we need to produce the architecture which produces the hand position as a 2D-vector by input hand image. Next the hand gesture class for the input image. By using the first method we can get 89% accuracy. By using the second method we can get 85.2% accuracy. In this paper we are using virtual method to avoid the original labelling image. By using this method we can get 30 different hand poses. This is also used in various areas like human pose estimation, driverless cars and mobile robotics.

Yanan Xu1, et al [19] in this paper is using the convex defection method. This method issuing the four different steps they are sample images capturing, image pre-processing, feature analysis and identification parameter extraction, classification and recognition. This method is used mainly due to the uniqueness and classification easily. This method also includes the tightness of gesture contour to its convex hull, namely, convexity, and the relative position of fingertips. This method is simple, efficient and free from gesture direction and position.

3. SUMMARY OF EXISTING WORK:

Table 1: Comparative analysis of the various methods

S.no	Author	Comparison of various techniques year	Image acquisition Method	No. of Gestures	Segmentation Technique	Features Extracted	Recognition Method	Accuracy
1	Nishihara, et al[1].	Jul. 4, 2017[1]	Stereo cameras-Image taken by two camera(first original image ,second image to check the original image)	22	Bressenham's midpoint circle scan-conversion algorithm	Hand Orientation	Hand-gesture recognition method	84%
2	Zhang, et al. [2]	Jan. 3, 2017[2]	Depth Sensor	10	Find Body from Blobs, Find Hand (Hypotheses) from Threshold, Refine by Removing Arm if Needed	Velocity (from Current - Previous Frame), Rotation from Saved Parameters, Shape Descriptor(s) including Dimensionality Reduction	Dynamic hand gesture recognition using depth data	94.40%
3	Quentin De Smedt, et. Al[4]	17 Jul 2017[4]	Intel Real Sense camera	14 and 28	Histogram oriented gradient(Confusion matrix)	3D depth information is used to extract hand silhouettes or simply hand areas in order to extract features from segmented hand region	1)Skeleton-based Dynamic hand gesture recognition 2)Classify sequence by key frames with convolutional neural network	88.24%(for 14 gesture) and 81.90(for 28 gesture)
4	Aashni Hariaa, et. Al[5]	24 August 2017[5]	webcam or USB based webcam	6-static,1-dynamic	Inverted Binary Thresholding and Otsu's Thresholding	Finger's using Contours extraction	Template Matching	around 92.2%
5	Yan et al.[6]	Jan. 24, 2017[6]	Hand gestures may be captured by a camera.	24 alphabets (upper and lower case) and 10	histogram	Fingertips using curvature and convexity defects, Euclidean distance, direction	perpendicular(exact shape of Surface along with direction) or convexity(when determining which alphanumeric)	86%

				numbers.				
6	Gang Li, Rui Zhang, et. Al[7]	May,2017[7]	Micro Doppler radar sensor(K-band CW radar)	4-dynamic	omp algorithm	sparsity-based feature extraction -Hand rotation, Calling, Snapping fingers, Flipping fingers	i)HMM for dynamic ii)SVM	85.16% to 91.46% finally (90%)
7	Zhuang, Huiwei, et al. [16]	Feb 22,2017	Non-negative matrix factorization (NMF) and compressive sensing (CS).	10	local binary pattern (LBP), histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT) and shape representations etc	We focus on the gesture recognition based on subspace analysis and we apply the NMF to get the subspace representation of gesture images.	static hand gesture recognition method	85.1-95.2%
8	A.O. Alghabri et al[17]	Oct(2017)	an ordinary 2D webcam	5	interactive multi applications AR system,	Recognition is performed based on shape features, whereas the depth feature, of gestures and markers, was estimated using an ordinary 2D webcam to reduce the power consumption and cost.	Argumented reality hand gesture recognition	93.20%
9	E I Nikolaev et al [18]	2018	web camera (with resolution 1024x768) and GeForce 750M graphics card	30	deeply-trained model for hand gesture recognition through the use of hand images	first architecture produces the hand position as a 2D-vector by input hand image. The second one predicts the hand gesture class for the input image.	virtual training of hand gesture	89.00%
10	Yanan Xu1, et al19]	Nov 12,2017	conture technique is used to detect the images	5	gradient histogram, image subspace projection, shape features, etc.	fingertip related features based on convex defect detection,	convex detection technique	85.00%

4. PROPOSED METHOD :

I. Capture frames and convert to grayscale:

Our ROI i.e. region of interest, is the hand region. Images of the hand are captured and converted to grayscale. We convert an image from RGB to grayscale and then to binary in order to find the ROI i.e. the portion of the image we are further interested for image processing. By doing this our decision becomes binary: "yes the pixel is of interest" or "no the pixel is not of interest".

II. Blur image:

Gaussian Blurring is used on the original image. We blur the image for smoothing and to reduce noise and details from the image. We are not interested in the details of the image but in the shape of the object to track. By blurring, we create smooth transition from one color to another and reduce the edge content. We use thresholding for image segmentation, to create binary images from grayscale images.

III. Thresholding

In very basic terms, thresholding is like a Low Pass Filter by allowing only particular color ranges to be highlighted as white while the other colors are suppressed by showing them as black.

Otsu's Binarization method is used. In this method, OpenCV automatically calculates/approximates the threshold value of a bimodal image from its image histogram. But for optimal results, we may need a clear background in front of the webcam which sometimes may not be possible.

IV. Draw contours

The next step is to find out contours of the image generated in the previous step. We know that no comparison method is 100% efficient and so no matter how perfect the histogram is or how efficient the back projection algorithm be, there will always be false detections and noise. So we don’t just find the contour of the image but also apply some methods to remove and false detection.

V. Find convex hull and convexity defects

We now find the convex points and the defect points. The convex points are generally, the tip of the fingers. But there is other convex point too. So, we find convexity defects, which is the deepest point of deviation on the contour. By this we can find the number of fingers extended and then we can perform different functions according to the number of fingers extended.

COMPARATIVE STUDY WITH SKIN COLOR DETECTION AND SEGMENTATION IN HSV AND YCBCR COLOR SPACE

In the above mentioned paper they have used the YCbCr color (Luminance, Chrominance) space one important task in image processing applications is the color space conversion. Real-time images and videos are stored in RGB color space, because it is based on the sensitivity of color detection cells in the human visual system. In digital image processing the YCbCr color space is often used in order to take advantage of the lower resolution capability of the human visual system for color with respect to luminosity. Thus, RGB to YCbCr conversion is widely used in image and video processing

In our project we are using the HSV color model there are numerous good reasons which one can understand by searching a bit on the internet. Basically we choose HSV because it is a better color space than RGB when it comes to detecting colors because Hue and Saturation channels are not affected by lighting and other image parameters and extracting color information is thus easier.

Digital image processing using computer vision technology is now widely used as a research object. Part of image processing is to use color based processing. Color analysis in the introduction of digital imagery there are several models including, RGB, CMY, HSI, HSV and normalized RGB models. One form of application of HSV model is as facial recognition. Using this model as facial recognition has the advantage of being simple in programming, the process is fast so it is perfect.

COMAPRATIVE STUDY OF TOF 3D-CAMERA AND WEBCAMERA:

In the above mentioned paper they have used the TOF- camera for the gesture recognition. Which is the advanced method when compared with all others method.this is more useful when compared to Nowadays, 3D data are required in the automation industries for analysing the visible space/environment. The rapid acquisition of 3D data by a robotic system for navigation and control applications is required. New 3D cameras at affordable prices which have been successfully developed using the ToF principle to resemble LIDAR scanners. In a ToF camera unit, a modulated light pulse is transmitted by the illumination source and the target distance is measured from the time taken by the pulse to reflect from the target back to the receiving unit. PMD technologies have developed 3D sensors using the ToF principle, which provide for a wide range of field applications with high integration and cost-effective production.

ToF cameras do not suffer from missing texture in the scene or bad lighting conditions, are computationally less expensive than stereo vision systems and - compared with laser scanners have higher frame rates and more compact sensors, advantages which make them ideally suited for 3D perception and motion reconstruction. 3D Time-of-Flight (TOF) technology is revolutionizing the machine vision industry by providing 3D imaging using a low-cost CMOS pixel array together with an active modulated light source. Compact construction, easy-of-use, together with high accuracy and frame-rate makes TOF cameras an attractive solution for a wide range of applications. In this article, we will cover the basics of TOF operation, and compare TOF with other 2D/3D vision technologies. Then various applications that benefit from TOF sensing, such as gesturing and 3D scanning and printing, are explored. Finally, resources that help readers get started with Texas Instruments' 3D TOF solution are provided.

DRAWBACK OF THIS SYSTEM:

- We can only show hand in the particular rectangular region.
- In Night light occurrence of system is less compared to day light.
- At long distance system fails to detect the hand.
- Multi hand detection is not possible.
- Only five gestures have been done in this document.
- Other than Region of interest object is recognized.

5. PYTHON CODE:

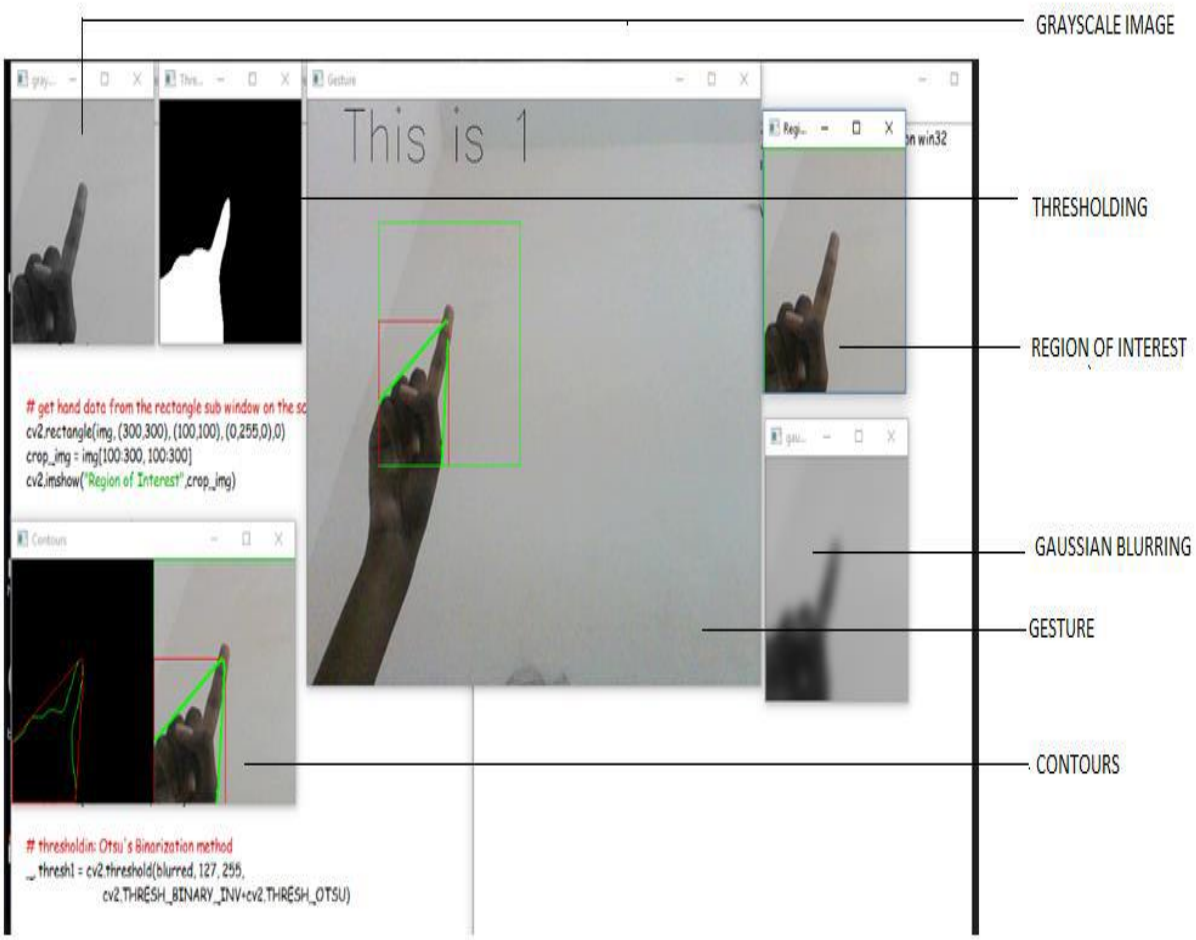
```
import cv2
import numpy as np
import math
cap = cv2.VideoCapture(0)
while(cap.isOpened()):
# read image
ret, img = cap.read()
# get hand data from the rectangle sub window on the screen
cv2.rectangle(img, (300,300), (100,100), (0,255,0),0)
crop_img = img[100:300, 100:300]
cv2.imshow("Region of Interest",crop_img)
# convert to grayscale
grey = cv2.cvtColor(crop_img, cv2.COLOR_BGR2GRAY)
cv2.imshow('grayscale image',grey)
# applying gaussian blur
value = (35, 35)
blurred = cv2.GaussianBlur(grey, value, 0)
cv2.imshow('gaussianblur',blurred)
# thresholdin: Otsu's Binarization method
_, thresh1 = cv2.threshold(blurred, 127, 255,
cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
# show thresholded image
cv2.imshow('Thresholded', thresh1)
# check OpenCV version to avoid unpacking error
(version, _, _) = cv2.__version__.split('.')
if version == '3':
image, contours, hierarchy = cv2.findContours(thresh1.copy(), \
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
elif version == '2':
contours, hierarchy = cv2.findContours(thresh1.copy(),cv2.RETR_TREE, \
cv2.CHAIN_APPROX_NONE)
# find contour with max area
```

```
cnt = max(contours, key = lambda x: cv2.contourArea(x))
# create bounding rectangle around the contour (can skip below two lines)
x, y, w, h = cv2.boundingRect(cnt)
cv2.rectangle(crop_img, (x, y), (x+w, y+h), (0, 0, 255), 0)
# finding convex hull
hull = cv2.convexHull(cnt)
# drawing contours
drawing = np.zeros(crop_img.shape,np.uint8)
cv2.drawContours(drawing, [cnt], 0, (0, 255, 0), 0)
cv2.drawContours(drawing, [hull], 0,(0, 0, 255), 0)
# finding convex hull
hull = cv2.convexHull(cnt, returnPoints=False)
# finding convexity defects
defects = cv2.convexityDefects(cnt, hull)
count_defects = 0
cv2.drawContours(thresh1, contours, -1, (0, 255, 0), 3)
# applying Cosine Rule to find angle for all defects (between fingers)
# with angle > 90 degrees and ignore defects
for i in range(defects.shape[0]):
s,e,f,d = defects[i,0]
start = tuple(cnt[s][0])
end = tuple(cnt[e][0])
far = tuple(cnt[f][0])
# find length of all sides of triangle
a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2)
b = math.sqrt((far[0] - start[0])**2 + (far[1] - start[1])**2)
c = math.sqrt((end[0] - far[0])**2 + (end[1] - far[1])**2)
# apply cosine rule here
angle = math.acos((b**2 + c**2 - a**2)/(2*b*c)) * 57
# ignore angles > 90 and highlight rest with red dots
if angle <= 90:
count_defects += 1
cv2.circle(crop_img, far, 1, [0,0,255], -1)
```

```
#dist = cv2.pointPolygonTest(cnt,far,True)
# draw a line from start to end i.e. the convex points (finger tips)
# (can skip this part)
cv2.line(crop_img,start, end, [0,255,0], 2)
#cv2.circle(crop_img,far,5,[0,0,255],-1)
# define actions required
if count_defects == 1:
cv2.putText(img,"This is V ", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
elif count_defects == 2:
str = "This is three"
cv2.putText(img, str, (5, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
elif count_defects == 3:
cv2.putText(img,"This is 4", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
elif count_defects == 4:
cv2.putText(img,"Hi!!", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
else:
cv2.putText(img,"This is 1", (50, 50),\
cv2.FONT_HERSHEY_SIMPLEX, 2, 2)
# show appropriate images in windows
cv2.imshow('Gesture', img)
all_img = np.hstack((drawing, crop_img))
cv2.imshow('Contours', all_img)
k = cv2.waitKey(10)
if k == 27:
break
```

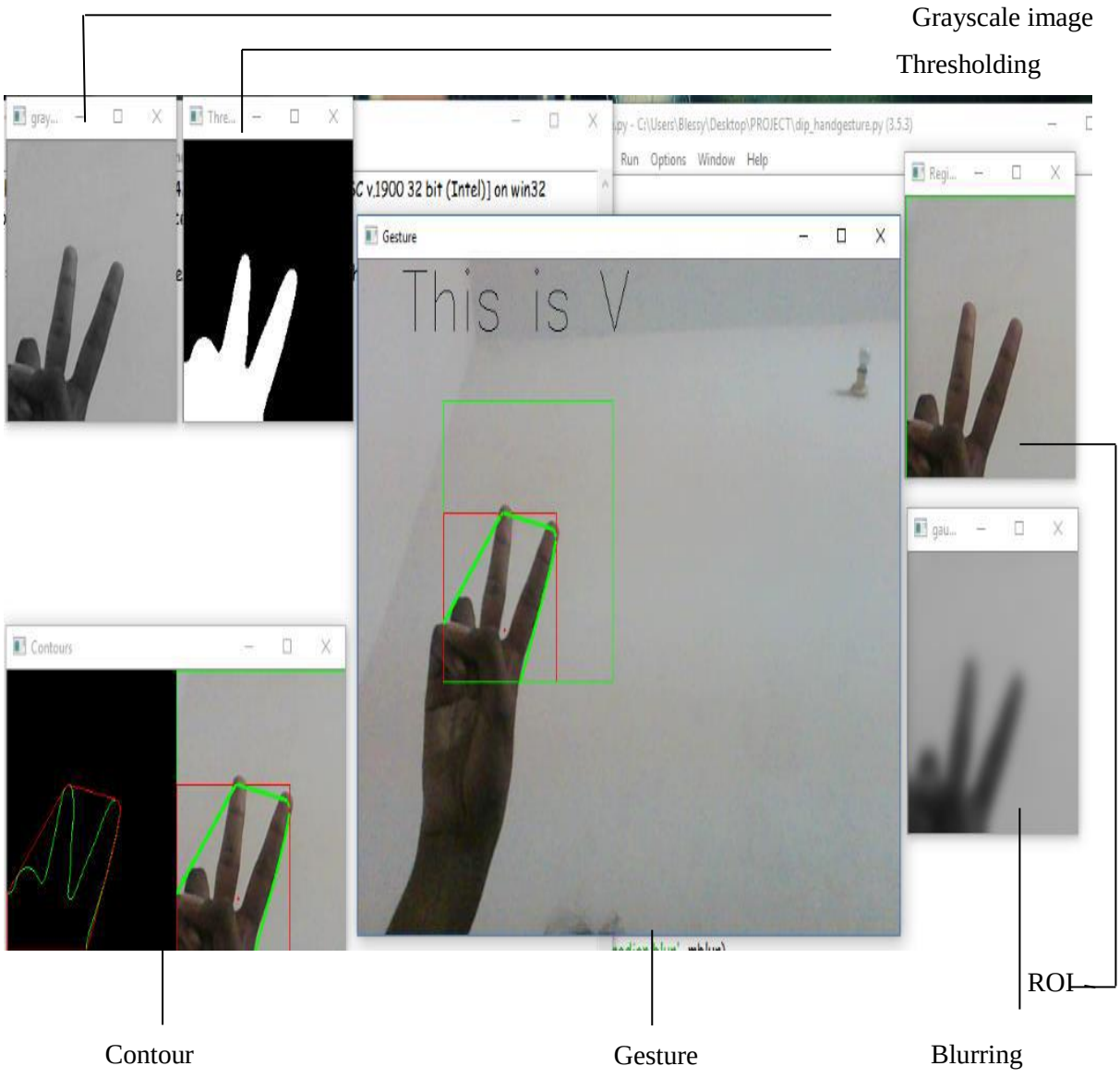
6. EXPERIMENTAL RESULT:

FOR "1" FINGER:



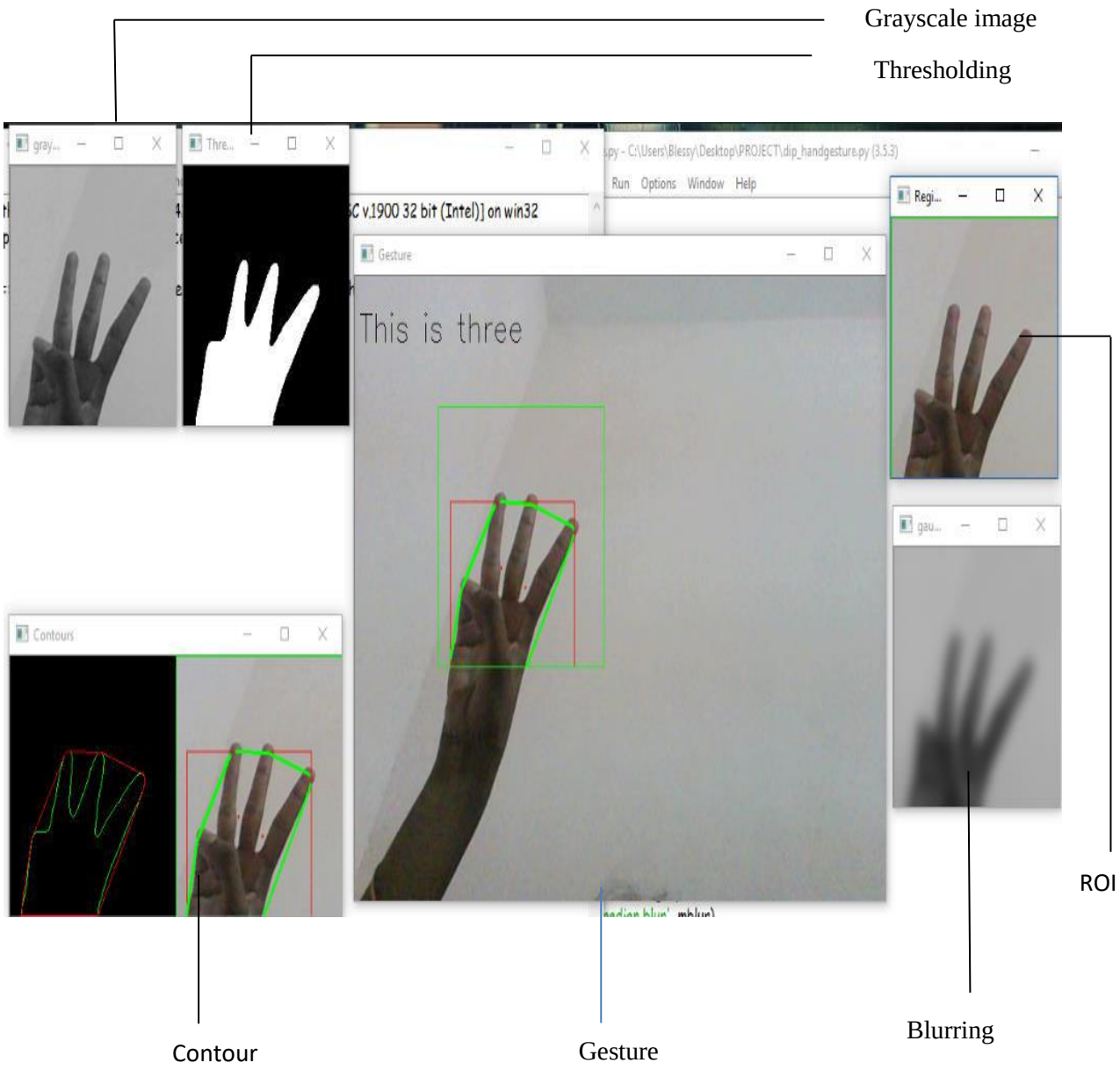
- From this output we get the image of our hand and this detected by one convex hull point hence this is display as ONE.
- Gray scale image, threshold image, contour image, Gaussian blurring, gesture, region of interest of hand is available in this output.

FOR "2" FINGERS:

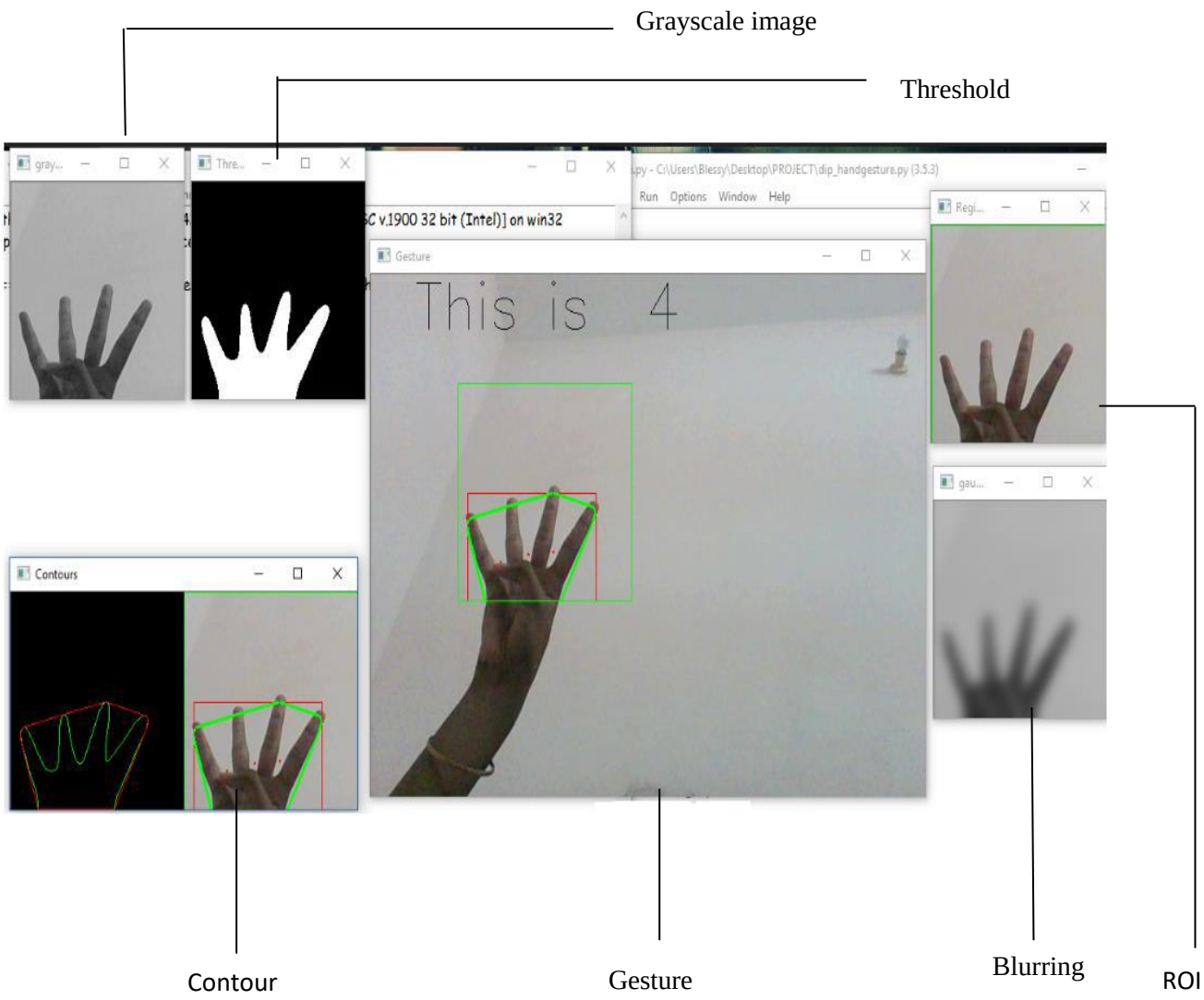


- This is the output of two fingers. In this there are 2 convex hulls. So this is displayed as V

FOR "3" FINGERS:

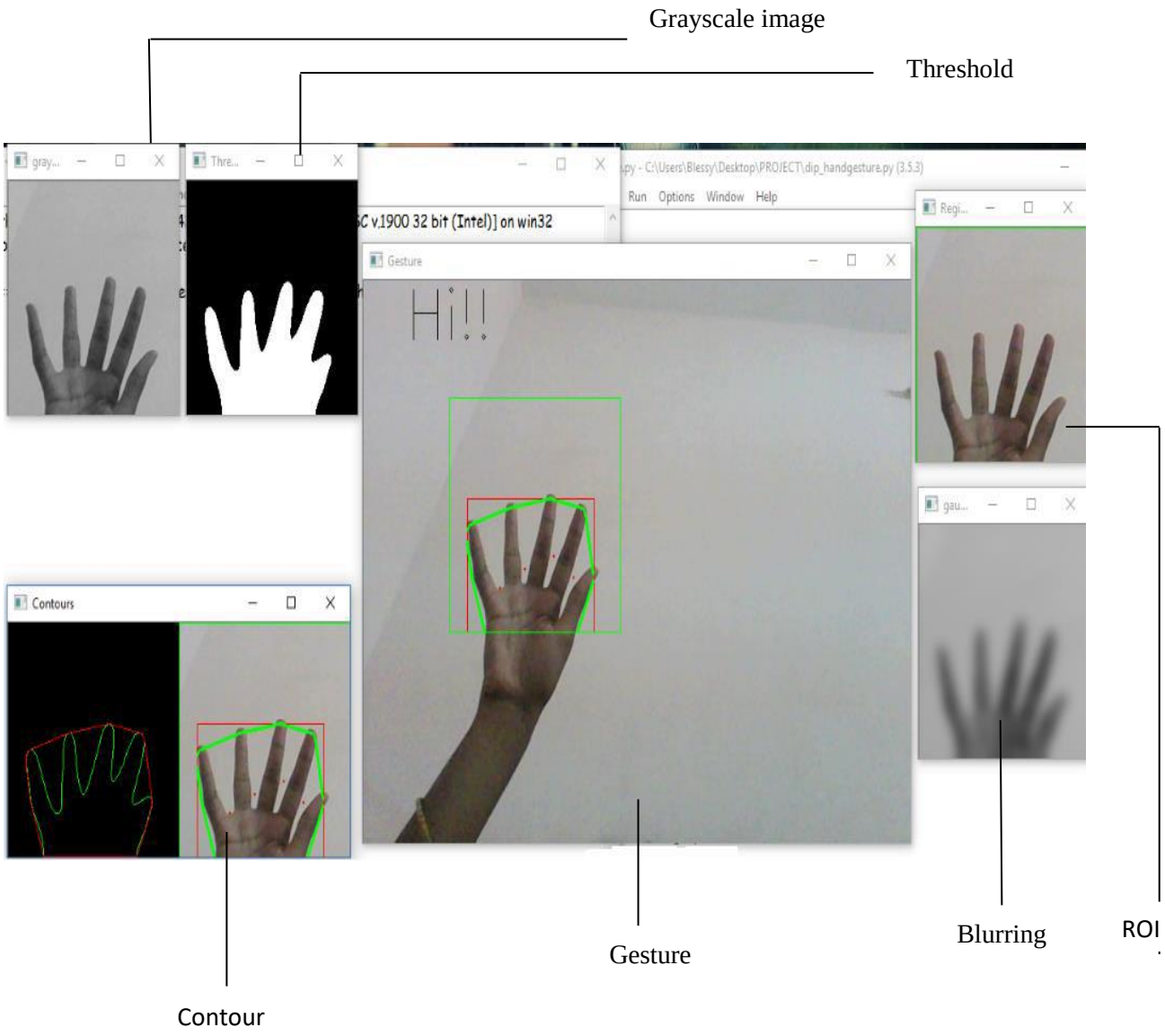


FOR "4" FINGERS:



- This is the output of four fingers. In this there are 4 convex hulls. So this is displayed as 4

FOR "5" FINGERS:



7. CONCLUSION:

In this paper the Hand Gesture Recognition methods proposed by several authors with different method have been studied. Through this study we can able to know about varies method used for hand gesture recognition. These methods have shown best performance for hand gesture recognition. Although the methods have proved the efficiency, there are some challenges which have to be addressed.

8. REFERENCE:

1) Nishihara, H. K., Hsu, S. P., Kaehler, A., & Jangaard, L. (2017). *U.S. Patent No. 9,696,808*. Washington, DC: U.S. Patent and Trademark Office.
2) Zhang, Z., & Kurakin, A. V. (2017). *U.S. Patent No. 9,536,135*. Washington, DC: U.S. Patent and Trademark Office.
3) Chen, D., Li, G., Sun, Y., Kong, J., Jiang, G., Tang, H., ... & Liu, H. (2017). An interactive image segmentation method in hand gesture recognition. *Sensors*, *17*(2), 253.
4) De Smedt, Q., Wannous, H., Vandeborre, J. P., Guerry, J., Le Saux, B., & Filliat, D. (2017, April). SHREC'17 Track: 3D Hand Gesture Recognition Using a Depth and Skeletal Dataset. In 10th Eurographics Workshop on 3D Object Retrieval.
5) Haria, A., Subramanian, A., Asokkumar, N., Poddar, S., & Nayak, J. S. (2017). Hand gesture recognition for human computer interaction. *Procedia Computer Science*, *115*, 367-374.
6) Yan, B., Weng, F., Ren, L., Cheng, Y. C., & Shen, Z. (2017). *U.S. Patent No. 9,551,590*. Washington, DC: U.S. Patent and Trademark Office.
7) Li, G., Zhang, R., Ritchie, M., & Griffiths, H. (2017, May). Sparsity-based dynamic hand gesture recognition using micro-Doppler signatures. In *Radar Conference (RadarConf), 2017 IEEE* (pp. 0928-0931). IEEE.
8) Nowozin, S., Kohli, P., & Shotton, J. D. J. (2017). *U.S. Patent No. 9,619,035*. Washington, DC: U.S. Patent and Trademark Office.
9) Amir, A., Taba, B., Berg, D. J., Melano, T., McKinstry, J. L., Di Nolfo, C., ... & Kusnitz, J. (2017, July). A Low Power, Fully Event-Based Gesture Recognition System. In *CVPR* (pp. 7388-7397).

10) Tsironi, E., Barros, P., Weber, C., & Wermter, S. (2017). An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. *Neurocomputing*, *268*, 76-86.
11) Mehra, V. (2018). Hand Gesture Recognition using Support Vector Machine and Bag of Visual Words model.
12) Marium, A., Rao, D., Crasta, D. R., Acharya, K., & D'Souza, R. (2017). Hand Gesture Recognition using Webcam. *American Journal of Intelligent Systems*, *7*(3), 90-94. (http://article.sapub.org/10.5923.j.ajis.20170703.11.html)
13) Liu, Z., Chai, X., Liu, Z., & Chen, X. (2017, October). Continuous gesture recognition with hand-oriented spatiotemporal feature. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3056-3064).
14) Patel, N. A., & Patel, S. J. (2018). Hand Gesture Recognition System For Human Computer Interaction (HCI).
15) Patel, N. A., & Patel, S. J. (2018). A Survey On Hand Gesture Recognition System For Human Computer Interaction (HCI).
16) Xu, Y., Park, D. W., & Pok, G. (2017). Hand Gesture Recognition Based on Convex Defect Detection. International Journal of Applied Engineering Research, 12(18), 7075-7079.
17) Nikolaev, E. I., Dvoryaninov, P. V., Lensky, Y. Y., & Drozdovsky, N. S. (2018, May). Using virtual data for training deep model for hand gesture recognition. In Journal of Physics: Conference Series (Vol. 1015, No. 4, p. 042045). IOP Publishing.
18) Alghabri, A. O., Osman, F. H., & Ahmed, N. Y. (2017). FPGA-Based Real Time Hand Gesture and AR Marker Recognition and Tracking for Multi Augmented Reality Applications. ARAB JOURNAL OF NUCLEAR SCIENCES AND APPLICATIONS, 50(3), 66-76.
19) Zhuang, H., Yang, M., Cui, Z., & Zheng, Q. (2017). A method for static hand gesture recognition based on non-negative matrix factorization and compressive sensing. IAENG International Journal of Computer Science, 44(1), 52-59.
20) Shaik, K. B., Ganesan, P., Kalist, V., Sathish, B. S., & Jenitha, J. M. M. (2015). Comparative study of skin color detection and segmentation in HSV and YCbCr color space. Procedia Computer Science, 57, 41-48.
21) Gesture Recognition: A Survey