Engineering and Technology Journal
Journal homepage: engtechjournal.org

Real-time Hand Gesture Extraction Using Python Programming Language Facilities

Azher A. Fahad [a]*, **Hassan J. Hassan** [b], **Salma H. Abdullah** [c]

[a] Computer Engineering Department, University of Technology, Baghdad, Iraq, azher003@yahoo.com

[b] Computer Engineering Department, University of Technology, Baghdad, Iraq ,60012@uotechnology.edu.iq

[c] Computer Engineering Department, University of Technology, Baghdad, Iraq, 120015@uotechnology.edu.iq

*Corresponding author.

Submitted: 24/02/2020 Accepted: 20/04/2020 Published: 25/06/2021

KEYWORDS

Hand Gesture,
Computer Vision,
Image Segmentation,
ASL Dataset.

ABSTRACT

Hand gesture recognition is one of communication in which used bodily behavior to transmit several messages. This paper aims to detect hand gestures with the mobile device camera and create a customize dataset that used in deep learning model training to recognize hand gestures. The real-time approach was used for all these objectives: the first step is hand area detection; the second step is hand area storing in a dataset form to use in the future for model training. A framework for human contact was put in place by studying pictures recorded by the camera. It was converted the RGB color space image to the greyscale, the blurring method is used for object noise removing efficaciously. To highlight the edges and curves of the hand, the thresholding method is used. And subtraction of complex background is applied to detect moving objects from a static camera. The objectives of the paper were reliable and favorable which helps deaf and dumb people interact with the environment through the sign language fully approved to extract hand movements. Python language as a programming manner to discover hand gestures. This work has an efficient hand gesture detection process to address the problem of framing from real-time video.

How to cite this article: A. A. Fahad, H. J. Hasan, and S. H. Abdullah, “Real Time Hand Gesture Extraction Using Python Programming Language Facilities,” Engineering and Technology Journal, Vol. 39, No. 06, pp. 1031-1040, 2021.
DOI: https://doi.org/10.30684/etj.v39i6.1619
This is an open access article under the CC BY 4.0 license http://creativecommons.org/licenses/by/4.0

1. INTRODUCTION

Hand gesture recognition is one active field of computer vision research. It offers to interact with devices without the additional device use [1]. The system belongs to human-computer interaction (HCI) techniques that enable the user to interact with the system without any difficulty. This technology can be used as an aid for handicap people. Body language is a substantial way of communication between humans[2], thus gesture extraction systems may be used for the beneficent Human-Machine Interface (HMI). This system interface would allow a Human consumer to remotely control a large variety of devices growing hand postures. The hand gesture recognition application domains [3] are sign language desktop applications, robotics, medical environment, home automation, virtual reality, smart TV, etc. Hand segmentation is the primary task of the process of gesture recognition.

The proposed framework, introduced in this paper, allows users to communicate with machines by hand postures, being the device under different backgrounds and lighting conditions. The paper focuses on the different stages involved in the detection of hand posture, from the image originally captured to customize dataset that will be used in proposed deep learning model training to achieve in the next paper as the final step, classification. This paper proposes a better way in which the background picture is taken at the beginning afterward the background picture is subtracted from the picture to detect the region of interest, which makes it easier to detect gestures. Such traditional input instruments are very user-friendly, easily accessible and easy to learn. But there is much less way for people with disabilities to interact with the machine today. This led to the development of a new kind of system which makes disabled people easily communicate with the system. The hand Gesture Recognition System will be the best means for disabled people to communicate with the system. Recognition of real-time gestures is a challenging task. Gestures are extracted from the video representing the static gesture performing signs. The use of information based on video is growing. In the field of video analysis, video description, video editing and animation, the mainframe are very useful.

2. RELATED WORK

Hand detection and removal of backgrounds are essential for the recognition of gestures. We need to isolate the region of the hand from the background so that the method for gesture recognition can work properly. Many methods of gesture recognition actually use the technique of adaptive thresholding [4 - 5]. Instead, [6 - 7] morphological operations refine identified regions.

Some gesture recognition methods [8] suggested background picture is taken at the beginning afterward to remove the background image from the image in order to detect the area of interest, making it easier to recognize gestures. Also, several methods [9 - 10] Implement object subtraction history by choosing the Otsu Process threshold. Subtraction methods [11, 12] in the background hold the camera fixed, take a background picture in advance and then extract the current image from the background image.

Upon hand detection, the image is transformed to black & white, i.e. the skin pixels are marked as white and non-skin pixels (i.e. background) are marked as black and then some preprocessing techniques [13] are applied such as image filling, morphological erosion to improve image quality and eliminate any noise and finally inversion operation is done that convert foreground(edges and curves) to black and background to white.

3. PROPOSED FRAMEWORK METHODOLOGY

In this paper, the four stages are used to achieve the objectives mentioned previously. These include the following: Data Acquisition, Image Pre-processing, Image segmentation, and Dataset. The block diagram of the proposed methodology is demonstrated in Figure 1.

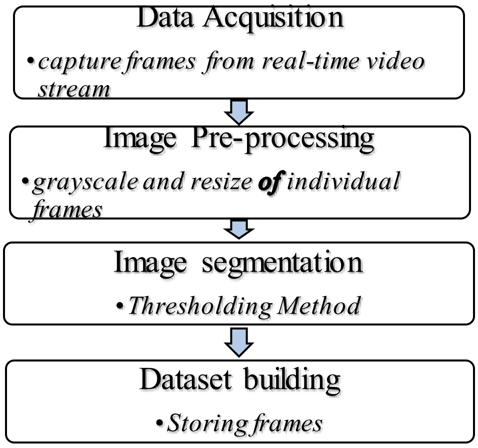


Figure 1: Block Diagram of the Proposed Methodology.

I. Data Acquisition

Firstly, capture a real-time video stream for hand gestures from a mobile device, that is Tablet, and get frames of the video for more calculations.

II. Image Pre-processing

Region of Interest (ROI), is segmented from the frame & resizes frame for 300 x 300 resolution. And then convert frames to grayscale. In conversion to grayscale, every image is a set of pixels. Pixels are the image's natural, building blocks. No finer granularity exists than the pixel. It usually thinks of a pixel as the color or intensity of light that appears in our image at a given location. The majority of pixels are shown in two ways: grayscale and color.

Pixel has a value between 0 and 255 in a grayscale image, where zero is "black" and 255 is "white". In the RGB color space, color pixels are usually represented – one value for the Red component one for Green, and one for Blue. In the range 0 to 255, each of the three colors is represented by an integer, which shows how much of the color there is. Such values are then combined in the form of an RGB tuple (red, green, blue).

Transformations in RGB space such as adding or removing the alpha stream, reverse the order of the channel, converting to or from the 8-bit of the RGB light, and converting to/from the grayscale using Eq. (1) [14].

$$RGB\ to\ Gray\ image\ A : Y = 0.299 * R + 0.587 * G + 0.114 * B \qquad (1)$$

III. Image segmentation

It is a method of partitioning a picture into separate regions with the same characteristics and is mostly used to isolate the area of interest (ROI). Segmentation creates a collection of homogeneous and significant areas so that pixels in each partitioned area share the same collection of properties or attributes. Sets of image properties can involve grayscale, contrast, spectral values, etc. [15].

The background segmentation method used to detach the target items from the background in capturing images with real-time video and computing the changes between frames. These process samples of previous images were saved in the memory and generate a background model based on the statistically valid characteristics of the samples. However, a binary image that works as a mask is built for the segmentation of background and objects. The background is divided into two parts, as follows: **Simple** Background, and **Complex** Background.

A. Simple Background

Based on a certain range of skin pixels, these mainframes are extracted from a video containing static hand gesture frames. To do that, we'll have to do some preprocessing of the image including Smoothing, Binarization, and Inversion operations. As demonstrated in Figure 2 that shows the Simple Background Method.

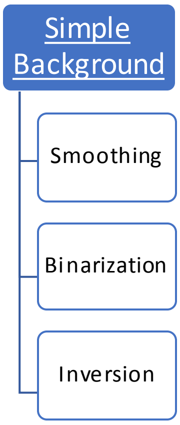
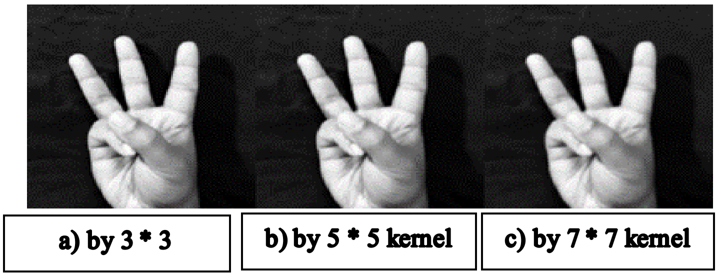


Figure 2: Simple Background Method.

The following subsections will discuss the approach in detail.

1) Smoothing is noise removing of the image by blurring. Blurring is what happens when the camera takes a picture out of focus. Sharper image regions lose their clarity, usually as a disk / circular shape. This means that each pixel of the image is combined with the pixel intensities of its surroundings [16]. This neighborhood mix of pixels is our blurred image. While this effect is typically undesirable in our images, when performing image processing tasks, it is actually quite helpful. Most image processing and computer vision functions such as thresholding and edge detection, work better when the image is smoothed or blurred first.
Various types of blurring methods exist **Gaussian**, **Averaging**, **Median**, and **Bilateral**.
In Gaussian blur, it is similar to blurring on mean, and rather than using a simple mean, instead, use a weighted mean where neighborhood pixels closest to the center pixel add more "weight" to the average. The result is that our picture is less blurred, but more realistic than the typical process as shown in Figure 3.



a) by 3 * 3 **b) by 5 * 5 kernel** **c) by 7 * 7 kernel**

Figure 3: Blurring of Image.

2) Binarization of the image & Inversion. Typically, thresholding is used for this process to highlight the edges of the hands using the Adaptive Thresholding- Otsu Method [17].
Thresholding is the image binarization. Generally speaking, trying to convert the grayscale picture to the binary model [15], where the pixels are either 0 or 255.
The threshold math is quite easy. If f(n) represents the pixel intensity of the input frame at that pixel coordinate, the threshold defines how accurate the image is to be displayed in a binary image, as seen below Eq. (2) [18].

$$f(n) = \begin{cases} 1, & if\ n\ >= threshold \\ 0, & if\ n\ < threshold \end{cases} \qquad (2)$$

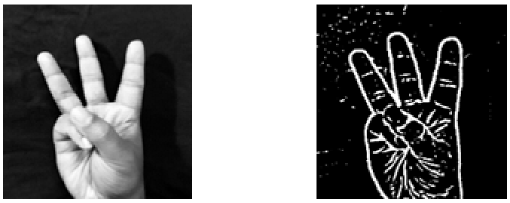
A simple example of thresholding would be to choose a pixel value v, then set all pixel intensities below v to zero, and all pixel values above v to 255. Creating a binary representation of the image in this way. Usually, using thresholding to focus on an image on objects or areas of special interest.
Various types of thresholding methods exist **Simple** thresholding, **Adaptive** thresholding, and Otsu's approach. We're going to look at the thresholding of Adaptive & Otsu.

By using adaptive thresholding that takes into account small pixel neighbors and then finds the optimum T threshold value for each neighbor. This approach enables us to handle cases where there may be significant pixel intensity ranges and T's optimal value can change for different parts of the image. The function transforms an image of a gray scale into a binary image using the formula for Adaptive thresholding as shown in Eq. (3) [19].

$$dsti(x,y) = \begin{cases} maxValue, & if\ sorc(x,y)\ > P(x,y) \\ \quad 0, & \qquad otherwise \end{cases} \qquad (3)$$

Where:

• sorc(x,y) – Image source or image of the input (1 stream, 8-bit or 32-bit floating-point).
• dsti(x,y) – The image of the destination is the same size and form as the sorc(x, y).
• maxValue – Value to be assigned to non-zero given to the pixels that are satisfied with the condition.
• P(x, y) – The level is determined independently of each pixel. As shown in Figure 4.



a) grayscale image **b) binary image**

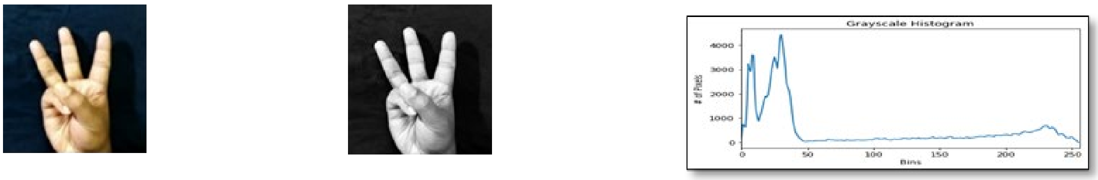
Figure 4: Adaptive thresholding of Image.

A histogram represents the pixel intensity (whether color or grayscale) distribution in a frame. This can be viewed as a graphic (or plot) that gives a high-level understanding of the intensity distribution (pixel quality). Having a general idea of differentiation, light, and strength distribution by simply analyzing an image's histogram.

By taking of a discrete greyish image x then let (ni) become the number of gray level (i) incidents. The chance of a pixel point (i) occurring in the frame is shown in the following Eq. (4) [20].

$$fx(j) = f(x = j) = \frac{nj}{n}, \qquad 0 \le j < M \tag{4}$$

Where M is the maximum number of grayscales in the image (usually 256), n is the maximum number of pixels in the image, as well as fx(j) is definitely the pixel value of the image histogram, which is standardized to [0, 1]. As shown in Figure 5.



a) Original image **b) grayscale image** **c) Histograms of gray scale**

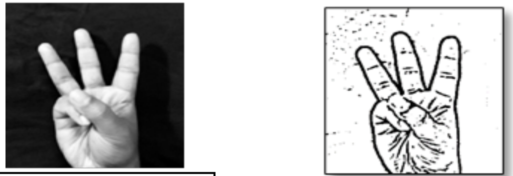
Figure 5: Grayscale histogram of the image.

The x-axis maps the bins (0-255). And the y-axis counts in each bin the number of pixels. The majority of the pixels fall in the range of roughly 0 to 50 and less from 50 to 255.

Otsu's approach assumes the image's grayscale histogram has two peaks[17]. But try to find an optimum value to distinguish such dual peaks – hence our T value.

Finally, Inversion Operation applied to convert foreground to black & background to white.

Otsu's method implementation of the image with inversion operation is shown in Figure 6.



a) The original grayscale **b) Implementing the Otsu approach**

Figure 6: Otsu's method implementation of image.

B. Complex background

Background subtraction is a method in image manipulation and computer vision Where the foreground image is collected for further processing (object recognition, etc.). In general, the regions of interest of an image are in the foreground objects (humans, vehicles, text, etc.). As shown in Figure 7.

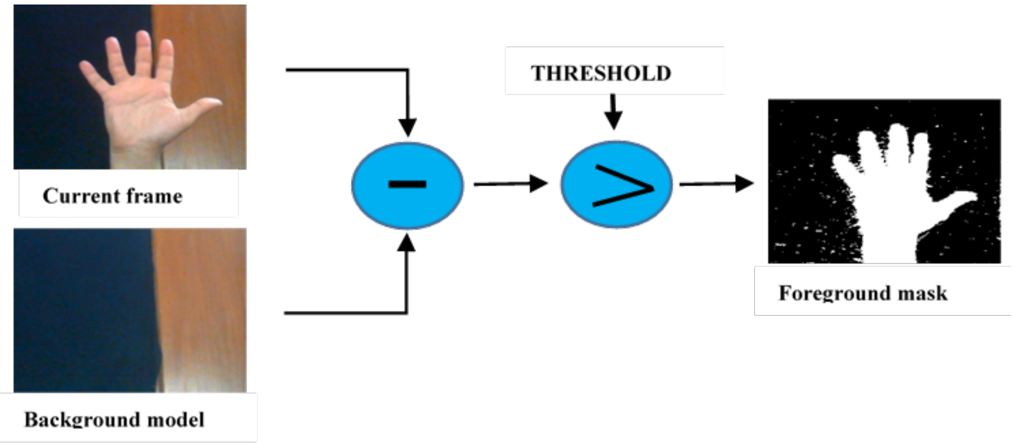


Figure 7: Complex Background Subtraction Concept.

Subtraction of the background is a commonly used technique to track moving objects in frames from live cameras. Background subtraction calculates the front mask by algebra between the frame buffer and the background image, including the stationary portion of the sequence or, in more general terms, everything that may be known as background due to the characteristics of the scene observed[8].

It includes the following steps: Background frame capture, Frame differencing, Image enhancement, Get a subset of an image, and finally Inversion. Figure 8 shows Complex Background Method.

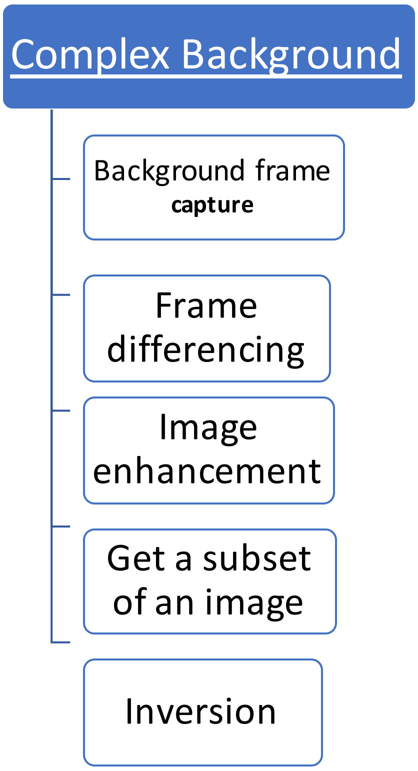


Figure 8: Complex Background Method.

The following subsections will discuss the approach in detail.

1) Background frame capture & Frame differencing

Take capture frame without a hand gesture, then take another with hand gesture and lastly makes frame subtraction (mask). The following equation is used to generate the foreground mask shown in Eq. (5) [21].

$$Bg(x,y,t) = Img(x,y,t-1) \quad (5)$$

Where: Bg(x, y, t): Background in time t, Img(x, y, t): Image in time t.

2) Image enhancement

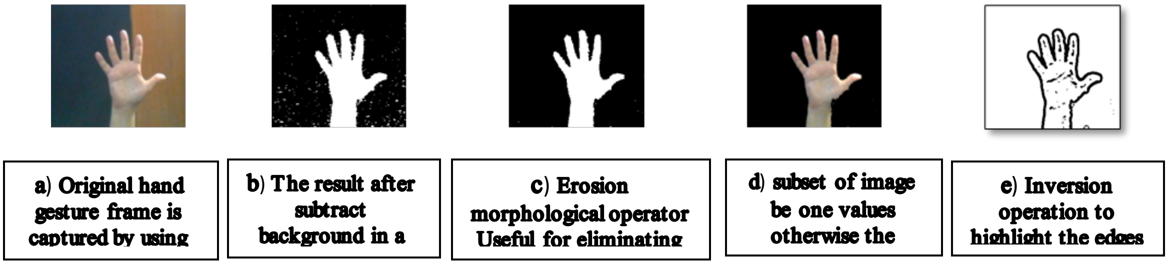
Using Morphological Transformation is a few basic image shape-based operations. It is usually done for binary screenshots. This requires two components, one of them is our original portrait, another is referred to as the structuring element or the kernel that determines the structure of the process. Using of Erosion morphological operator [13] that is just like soil erosion itself, it erodes the foreground object boundaries (always seek to keep the foreground white). Useful for eliminating tiny white noises.

3) Get a subset of an image

It defined by another image (mask) to get hand gestures by bitwise operations. By using bitwise and a function that computes the bit-wise each-element combination of two arrays (the frame that is taken from camera & mask that is extracted from cv2.createBackgroundSubtractorMOG2 function) or an array and a scalar. "And" operation will only be performed if the mask is not equal to zero, otherwise the result will be zero. With a single channel, the mask should be either white or black.

4) An inversion operation

to convert foreground to black & background to white. Subtraction of the complex background method is shown in Figure 9.



a) Original hand gesture frame is captured by using

b) The result after subtract background in a

c) Erosion morphological operator Useful for eliminating

d) subset of image be one values otherwise the

e) Inversion operation to highlight the edges

Figure 9: Subtraction of the complex background.

IV. Dataset Building

The human hand gesture images are taken for 26 alphabet signs of ASL,10 Counting numbers, and 3 special characters for three different persons. For each sign, there are 1500 images and for each person. There are 3x1500 x (26 alphabets + 10 numbers + 3 characters) images.

After that, frames were stored in folders as images. The name of the folder is used to label the images according to the type of gestures in the frame (i.e. label A for alphabet A and so on for other gesture types). It is possible to use any sign language representation, the representation proposed is for the American Sign Language Alphabet, see Figures 10 & 11.

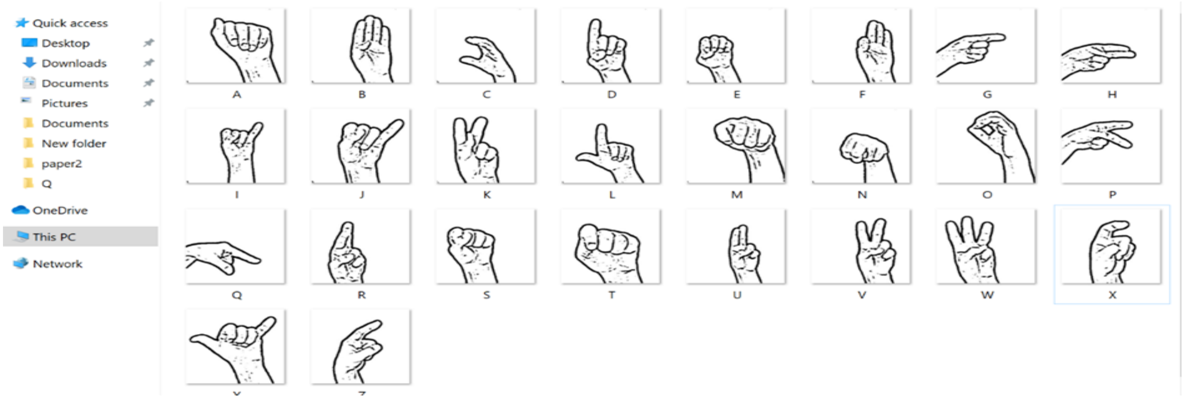


Figure 10: Sample of images of our dataset

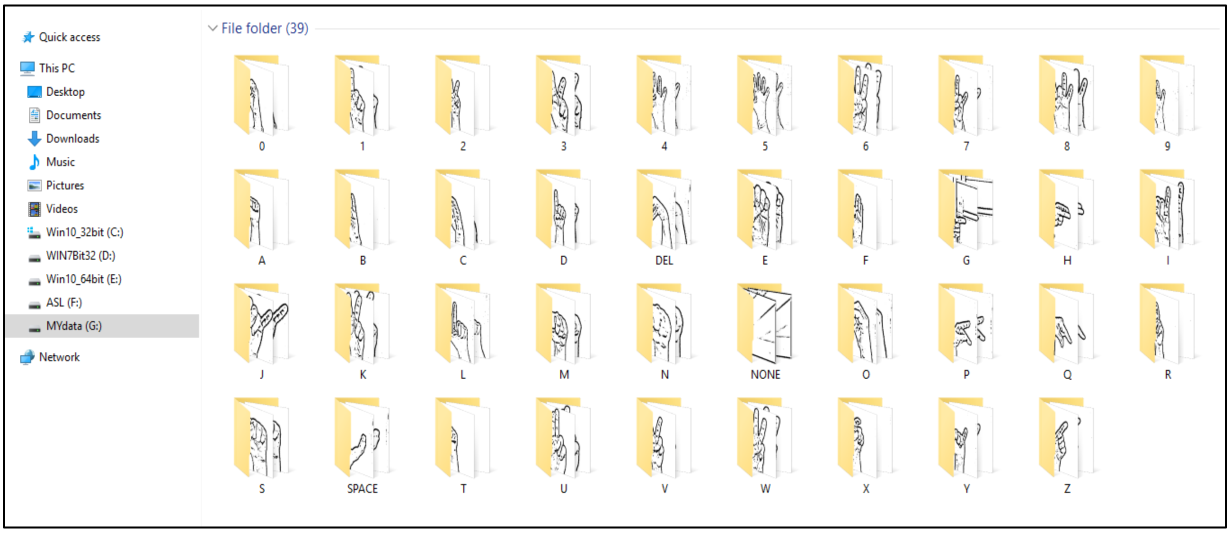


Figure 11: Dataset images in folders (American Sign Language)

4. Dataset Comparison with Other

Reza Azad et al. [22] had created their database for each character of ASL, which can include 504 images i.e. 21 images for each (24) gesture. While we created a dataset with more images and more gesture type i.e. 1500 images for (40) gestures as shown sample of them in Figures 10 and 11 that increase the accuracy of the classifier that will construct in the next paper because it's based on the deep learning concept when data increase, the result be more accurate. Also, during creating database images they proposed that captured should have a uniform dark color background that can be black with a white color rubber glove on hand as in contrast. They had done this in order to minimize noise and unwanted data so that they can easily do the segmentation process. While we tolerate uniform dark color because we can use subtract complex background manner to do the same about uniform dark color for the background. They propose that the user had to wear a black colored cloth around his arm till wrist from the shoulder so that the black color cloth can easily match with the background. The covered arm and the background should be of similar color. While in our work we didn't do it. Also, they proposed Letter j and z are discarded because they could describe them dynamically only and their approach is for static gestures only. While we proposed them as static gestures.

5. Result

This section is dedicated to evaluating the implemented method. The aims of our work are to implement a real-time hand gesture detection method and built a customized dataset from extracted gestures from a live camera. Image Acquisition is the first stage in the work to get frames from a real-time video stream by linking a mobile device camera to Python that allows getting frames from a live camera. Different processing techniques are applied to the frames after it has been getting in order to detect the hand gesture. After detection of hand gesture, the next step is storing the frames as images in folders which are represent customized dataset that will be used to train the proposed deep learning model in the next paper to achieve the classification step as a final result to build a system of hand gesture recognition in real-time. Our customized dataset consisting of 40 signs of three different persons. Every sign comprises 1500 images for each individual. So, in all, there are (3 x 1500 x 40) images. In the simple background, the result was excellent, but for complex background, the method suffers from noise because of lighting variations. In other words, when implementing the detection framework using background subtraction, we encountered several drawbacks and accuracy issues. Background subtraction cannot deal with sudden, drastic lighting changes leading to several inconsistencies. So, we preferred to make detection in a uniform background or using a USB LED Light in dimly lit conditions.

6. CONCLUSION

In this paper, we were able to create a robust gesture detection framework. Hence making it more user-friendly and lower cost. The work focuses on hand gesture detection based on adaptive thresholding to extract edges and curves of the hand. First, the Otsu method has been utilized to detect the hand gesture in a real-time video sequence. Second to build a customized dataset we adopted ASL in represented our gesture type in the customized dataset. This approach will be helpful to a sign language recognition system that will be achieved in the next paper. The experiment of this approach is to use for the static gesture. Extended works of this approach are detecting and tracking dynamic hand gestures.

References

[1] S. Anwar, S. K. Sinha, S. Vivek, V. Ashank, Hand gesture recognition: A survey, Lect. Notes Electr. Eng., 511 (2018) 365–371. https://doi.org/10.1007/978-981-13-0776-8_33

[2] S. Pramada, D. Saylee, N. Pranita, N. Samiksha, A. S. Vaidya, Intelligent Sign Language Recognition Using Image Processing, IOSR J. Eng., 3 (2013) 45–51. https://doi.org/10.9790/3021-03224551

[3] M. M. Hasan, P. K. Mishra, Hand Gesture Modeling and Recognition using Geometric Features : A Review, Canadian J. Image Process. Comput. Vis., 3 (2012) 12–26.

[4] O. P. Verma, P. Singhal, S. Garg, D. S. Chauhan, Edge detection using adaptive thresholding and ant colony optimization, 2011 World Congr. Inf. Commun. Technol, Mumbai, India, (2011), 313–318. https://doi.org/10.1109/WICT.2011.6141264

[5] M. H. Rahman, M. R. Islam, Segmentation of color image using adaptive thresholding and masking with watershed algorithm, 2013 Int. Conf. Informatics, Electron. Vision, Dhaka, Bangladesh, (2013) 1–6. https://doi.org/10.1109/ICIEV.2013.6572557

[6] Bosubabu Sambana, Internet of Things: Applications and Future Trends, Int. J. Innov. Res. Comput. Commun. Eng., 5 (2017) 5194–5202.

[7] H.N. Abdullah, H. K. Abduljaleel, Deep CNN Based Skin Lesion Image Denoising and Segmentation using Active Contour Method, Eng. Technol. J., 37 (2019) 464–469. http://dx.doi.org/10.30684/etj.37.11A.3

[8] K.Kavitha, A.Tejaswini, Background Detection and Subtraction for Image Sequences in Video, Int. J. Comput. Sci. Inf. Technol., 3 (2012) 5223-5226 .

[9] S. Shivashankara, S. Srinath, Palm extraction in american sign language gestures using segmentation and skin region detection, Int. J. Innov. Technol. Explor. Eng., 8 (2019) 2409–2418.

[10] R. K. Kumar, L. J. A. Jacob, P. Ayyanar, Gesture Based Real Time Security System Using Otsu Algorithm, Int. J. Adv. Sci. Eng. Technol., 1 (2013) 91–95.

[11] H. Lee, H. Kim, J. I. Kim, Background Subtraction Using Background Sets with Image-and Color-Space Reduction, IEEE Trans. Multimed., 18 (2016) 2093–2103. https://doi.org/10.1109/TMM.2016.2595262

[12] R. F. Rahmat, T. Chairunnisa, D. Gunawan, M. F. Pasha, R. Budiarto, Hand gestures recognition with improved skin color segmentation in human-computer interaction applications, J. Theor. Appl. Inf. Technol., 97 (2019) 727–739.

[13] A. M. Raid, W. M. Khedr, M. A. El-dosuky, M. Aoud, Image Restoration Based on Morphological Operations, Int. J. Comput. Sci. Eng. Inf. Technol., 4 (2014) 9–21. https://doi.org/10.5121/ijcseit.2014.4302

[14] L. Velho, A. C. Frery, J. Gomes, Image Processing for Computer Graphics and Vision, 2nd edition, London: Springer, 2009.

[15] P. Shashi, R. Suchithra, Review Study On Digital Image Processing And Segmentation,Am. J. Comput. Sci. Technol., 2 (2019) 68-72. https://doi.org/10.11648/j.ajcst.20190204.14

[16] M. Sonka, V. Hlavac,R. Boyle, Image Processing, Analysis, and Machine Vision, 3rd edition, Canada, 2008.

[17] P. Roy, S. Dutta, N. Dey, G. Dey, S. Chakraborty, R. Ray, Adaptive thresholding: A comparative study, Int. Conf. Control. Instrumentation, Commun. Comput. Technol., (2014) 1182–1186. https://doi.org/10.1109/ICCICCT.2014.6993140

[18] D. Kaur, Y. Kaur, Various Image Segmentation Techniques: A Review, Int. J. Comput. Sci. Mob. Comput., 3 (2014) 809–814.

[19] K. Bhargavi, S. Jyothi, A Survey on Threshold Based Segmentation Technique in Image Processing, Int. J. Innov. Res. Dev., 3 (2014) 234–239.

[20] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis, and Machine Vision, 4th edition, Stamford: Cengage Learning, 2014.

[21] P.-G. Ho, Image Segmentation, First Edit. London: InTech, 2011.

[22] R. Azad, B. Azad, I. T. Kazerooni, Real-time and robust method for hand gesture recognition system Based on cross-correlation coefficient, Adv. Comput. Sci. Int. J., 2 (2013) 121–125.