



Static Hand Gesture Angle Recognition via Aggregated Channel Features (ACF) Detector

Nabeel M. Mirza^{1*}, Duaa A. Taban², Ali J. Karam³, Anwar H. Al-Saleh⁴, Ali A. Al-Zuky⁵

¹ Department of Physics, College of Education, Mustansiriyah University, Baghdad 10052, Iraq

² Department of Physics, College of Education for Pure Science (Ibn Al- Haitham), University of Baghdad, Baghdad 10053, Iraq

³ Department of Information Technology, College of Computer and Information Technology, University of Garmian, Kurdistan Region 32007, Iraq

⁴ Department of Computer Science, College of Science, Mustansiriyah University, Baghdad 10052, Iraq

⁵ Department of Physics, College of Science, Mustansiriyah University, Baghdad 10052, Iraq

Corresponding Author Email: nabeel.mirza@uomustansiriyah.edu.iq

<https://doi.org/10.18280/ts.390320>

ABSTRACT

Received: 28 February 2022

Accepted: 12 May 2022

Keywords:

static hand gesture, image labelling, aggregate channel features (ACF) detector, AdaBoost algorithm, hand angle detection

Static hand gesture recognition is critical in the development of a system for human-computer interaction. Many human-computer interactions, such as human-robot interaction, game control, control of smart home devices, and others, use hand gestures as a fundamental and natural language of the body. The direction of rotation of static hand gestures is the subject of this research, and the focus is on six degrees of rotation (0°, 45°, 90°, 180°, 270°, and 315°). This work presents an ideal approach that can recognize the angle of hand gestures based on the Aggregate Channel Features (ACF) detector. This approach consists of three main stages: preprocessing (image labelling), computer training, and hand angle detection based on the ACF detector. The training process consists of 25 stages. The static hand gesture dataset contained 569 images (361 for training and 208 for testing). The average time cost to detect all hand gesture angles was 0.9445 seconds, and all hand angles were recognized with 100% accuracy. This is a strong indication that supports our approach.

1. INTRODUCTION

Studies regarding hand and face gesture recognition began at the end of the twentieth century and have continued since then. The results of those studies are what have resulted in various applications in several fields, such as robots [1], remote control, virtual reality, human computer interaction [2], controlling drones [3], etc. Remote control systems are widely used in the contemporary world, whether at home or at work, and they significantly aid us in our everyday activities [1]. Television and receivers are among the most common applications for remote control. However, the device's applications grew, and it has been incorporated into a variety of everyday applications, including surveillance devices, drones, and video game consoles.

In the future, human-computer interfaces are likely to allow more realistic, intuitive contact between people and all types of sensor-based devices, emulating interpersonal interactions more closely [4]. A gesture recognition system must be able to work in complex scenes with diverse backgrounds under varying lighting conditions in order to be real-time, stable, and deployable in unregulated environments [5]. There are two different types of gestures: dynamic gestures, in which the arm with the hand moves, and postures that change depending on the time interval [6]. In the second type, the body posture or arm does not change; just the hand stays in the same stance across time. This is known as a "static gesture" [7].

Hand gesture detection is achieved using vision techniques, which are more natural and do not require any gadgets or sensors that obstruct hand movement [8]. Different types of

cameras were utilized as sensors in vision methods to detect hand motions. Different techniques for detecting hand gestures using images collected from cameras have been developed [9]. These algorithms recognize hands (objects) by extracting visual information such as skin color [10], shape, motion, skeleton [9] or a combination of them that may be linked to the existence of hands in the camera's field of vision. Viola-jones [11] and Aggregate Channel Characteristics (ACF) object detector methods [12] are examples of machine learning techniques that employ the AdaBoost algorithm to extract object (hand) features.

Boosting algorithms [13] combine numerous models (weak learners) to obtain the final result (powerful learners). Adaptive boosting (AdaBoost) aims to concentrate on forecasting the current data set by giving each feature equal weight. In the event that the forecast is inaccurate, the inaccurate observation is given more weight [14]. The process is repeated until there are nearly no errors. A fundamental algorithm or decision stump is used to make the basic prediction. The weighted average of each weak classifier is used to forecast the final output. This technique is adaptive in nature since the weights are generated based on the inaccurate predictions and altered appropriately, so it is called AdaBoost. The approach employs a training collection of images that includes both negative and positive instances, such as faces and non-faces, hands and non-hands, and other objects.

The Image Labeler is a tool that labels regions of interest using bounding boxes. The Image Labeler provides a table with two columns for positive samples: the first column contains data names, and the second is an M-by-4 matrix for

the object's bounding box, which contains information about the object's position and size. Providing a set of negative samples for the technique to automatically create negative samples, as well as modifying the number of stages, feature type, and other function parameters, is required to achieve the desired detector accuracy.

Image labels, as stated in the study of Sager et al. [15], assist users in image labeling and achieve a variety of aims: (a) Labeling a huge number of images in a short period of time. (b) Assisting in the establishment of a full and balanced image collection. (c) Improving labeling efficiency and thereby lowering the human burden. (d) Maintaining high accuracy while eliminating label noise. (e) Encoding the extracted knowledge in a systematic and efficient manner.

The Aggregated Channel Features (ACF) detector employs a boosted tree to aggregate 10 channels, and the channels used are the same [8]: (3 channels) for LUV color space, (6 channels) for histogram of oriented gradient (HoG), and a normalized gradient (1 channel). The ACF object detector recognizes specific objects in images based on the training images and the object ground truth locations used with the trained object detector function.

In this work, a novel method based on an ACF detector is used to recognize six directions of the hand. The gestures vary according to the angle at which the hand rotates within the plane of the image. The current method for detecting hand direction has three stages: image labelling, computer training on a collection of 361 images, and then the ACF algorithm is used to detect the rotation angle (direction) of the hand. After computer training, a MATLAB training file containing the hand features is created, which is used in the suggested algorithm to detect the hand angle of rotation. The detection score acquired from the training file is used to estimate an angle of the hand (direction).

The primary contributions of this work: (1) proposing a method for determining the direction of hand gestures based on the detection confidence score (S_c). (2) Estimating the average time required to detect hand gesture angles.

The following is how the last four sections of the paper are organized. The second section covers modern approaches to detecting objects in images. Section 3 describes our method for recognizing hand gesture angles. Section 4 presents and discusses the results. The conclusion is finally in section 5.

2. RELATED WORK

There are two categories of techniques for identifying objects in an image: deep learning approaches and non-deep learning object detectors [16]. Viola and Jones [17] and ACF [12] are examples of non-deep learning techniques. The issue with detecting an object in an image is obtaining a region, suggesting, and testing it according to the class of object. As a result, some researchers relied on a single approach to identify the object in the image as in the studies [18-20]. While others used a mix of two approaches as in the studies [21-24]. Several recent research on object detection in an image has revealed that large-scale machine learning and deep learning techniques have proven impressive results in object recognition challenges [23, 25-27]. Therefore, many researchers have begun to apply machine-learning approaches to the challenge of detecting hand gestures. For example, Jonietz et al. [28], who used a positive dataset from three publicly accessible hand gesture datasets to detect palm hand gestures using ACF

machine learning. Investigation of pedestrian detection using Convolutional Neural Networks (CNNs) based on ACF feature extraction by Ghorban et al. [29]. Hermawati et al. [22] used a combination of the ACF detector and the Faster Convolutional Neural Network (CNN) R-CNN and were able to detect a cross-sectional region of the fetal limb in an ultrasound image. Admasu and Raimond [30] suggested a method for recognizing static hand gestures. Preprocessing, feature extraction, training, and identification are the three main processes in their system. They used the PCA method to extract features from each hand's sign and the ANN for recognition. de Smedt [31] suggested a system for identifying dynamic hand gestures relies on three gestural elements derived from hand skeleton sequences (hand direction, rotation, and a hand shape descriptor). He encoded each set of statistically extracted features using the Fisher Kernel. In the classification process, he used a linear SVM. The Deconvolutional Single Shot Detector (DSSD) approach applied in the study of Zhang et al. [20] to increase the detection accuracy and speed for identifying real-time static gestures.

3. METHODOLOGY

Four key elements of our approach discussed in this part are data acquisition, image labelling, data training, and hand angle recognition. Figure 1 depicts the scheme of our methodology.

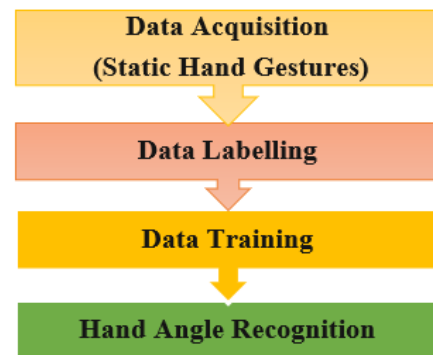


Figure 1. Basic procedure for detecting hand angle

3.1 Data acquisition

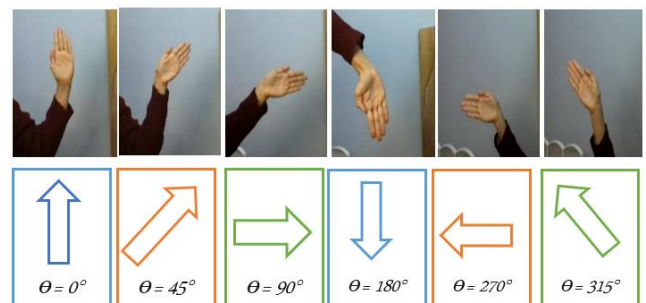


Figure 2. Samples of the data images

The positive image dataset consists of 569 images of the hand (palm posture) captured using the built-in webcam of a computer with a resolution of 1280x720 pixels. These images were captured with different angles, under fixed lighting conditions (283 Lux), and at distance (0.9 m) from the camera.

The distance may also be modified by zooming in or out depending on the camera's quality, so it is not required to determine the exact position of the palm in the image, but it is important to decide the direction of the palm at specific angles. Figure 2 manifests the samples collected at their respective angles.

3.2 Data training

To guarantee that the training process is proper and dependable, we first arrange all of the positive data and rotate them with a constant directional angle (upwards). Therefore, the original image of the hand is rotated at an angle of (I_{45}) by (-45) such that the palm of the hand faces up. Similarly, the images of the hand at different angles (I_{90} , I_{180} , I_{270} , and I_{315}) are rotated by (-90), (-180), (+90), and (+45), respectively. Figure 3 shows the outcome of this step.



Figure 3. Rotating the hand direction up

3.3 Image labelling

When training a model to detect an object, every instance of that object in the images must be labeled. There will be false negatives on the detection model if an object in an image is unlabeled. In this work, ACF Object Detector command is used to create an ACF object detector that can detect hand angle. The ACF detector uses a bounding box will extract the ROI, consisting of the positive region (hand), by the Image Label Toolbox (MATLAB), manually.

The ROI information for the positive data (hand: location and size of a hand) is arranged in a table and kept. After that, the (ACF) classifier trains on the position and size of the hand to create a model that will be employed later in the hand angle detection test. As a result, a two-column table will be constructed, one with image file names and the other with an M-by-4 matrix of M bounding boxes that circle the hand. Each bounding box uses the format [x, y, Width, Height] to indicate the location and size of the hand.

ACF Object Detector used to train computer using training data of hand in positive images dataset and negative image. Where, the negative images are images without a hand that are automatically recognized during training from the background of the positive image collection.

According to Viola and Jones research [11, 17], increasing the number of stages increases the number of features, thereby allowing the focus to be reduced the error detection rate and thus achieving a low false positive ratio. Therefore, in this work, the adaptive boosting (AdaBoost) classifier algorithm was adopted to implement the training in 25 stages (default options available in MATLAB). Each stage of classification determines whether the image contains a hand or not.

The suggested training strategy is based on one condition for all variables in the data, ensuring that the hand direction is always upward, and therefore a trained model is created using the AFC for one condition (the hand direction is up). The trained ACF file is returned as a MATLAB file and saved as a post-training with an extension of (.mat). This file will be used to detect hands later. Figure 4 shows a block diagram of the

training technique.

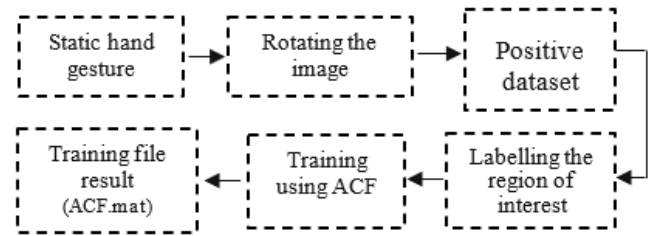


Figure 4. Main steps of proposed training

3.4 Hand angle detection

A novel strategy is used to detect hand angles based on the MATLAB file created during the training step (ACF.mat). The first step of the procedure, as shown in Figure 5, involves rotating an entered image (I_{in}) with a given angle into six new angles (0° , 45° , 90° , 180° , 270° , and 315°); as a consequence of this process, each image yields six new images (I_0 , I_{45} , I_{90} , I_{180} , I_{270} , and I_{315}) as shown in Figure 6.

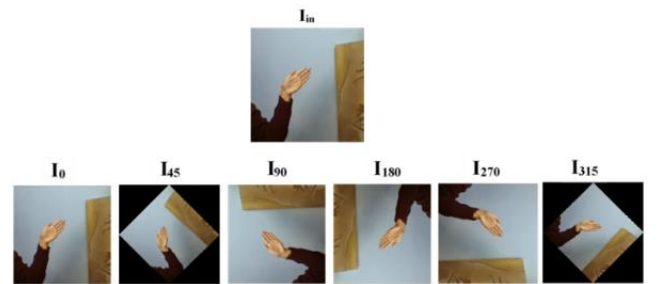


Figure 5. The positive image with different rotation angles (0° , 45° , 90° , 180° , 270° , and 315°)

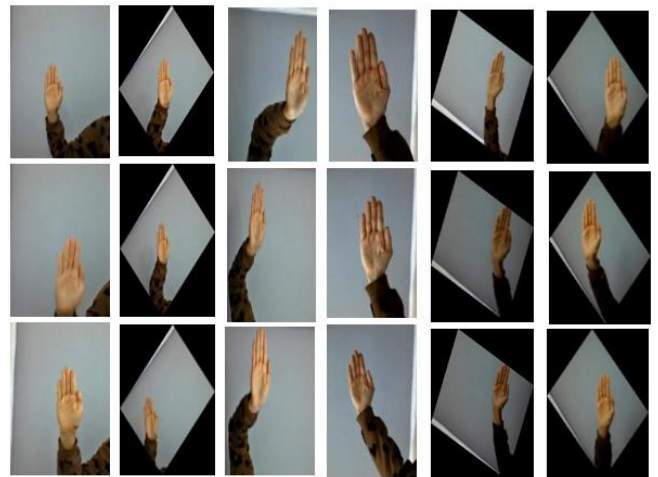


Figure 6. Rotation of data training

The second step is applying the training file (ACF.mat) to each rotated image. Bounding Box (bbox) is represented by a four-element vector [x, y, width, and height] as a rectangle enclosed by the detected hand to determine its location and size. Each hand's bbox has a detection score (S_c), which represents confidence in the detection. A higher score (S_c) value indicates a higher detection confidence. Since each input image (I_{in}) actually rotated to six different angles (I_0 , I_{45} , I_{90} , I_{180} , I_{270} , and I_{315}) during the testing process, Therefore, each

input image will have 6 bbox values, one value is chosen based on the maximum Sc value.

Maximum Sc denotes a specific rotated image at the angle (tht) that determines the input image angle. Whereas, a single maximum value of Sc (one rotating image) is determined from the images I_0 , I_{45} , I_{90} , I_{180} , I_{270} , and I_{315} . For example, if the larger score corresponds to the image I_{45} (an image rotated at an angle of 45°), then the angle of the input image is 45° . Consequently, the angle of the hand within the image plane is determined based on the score of detection (Sc). For further information, see the stages given below, which summarize the suggested approach.

Input: image I_{in}

Input: $\theta = (0^\circ, 45^\circ, 90^\circ, 180^\circ, 270^\circ, \text{ and } 315^\circ)$

Input: training file (ACF.mat)

Output: angle of input image (θ)

Step 1: rotate I_{in} according to (θ)

Step 2: applying (ACF.mat) on rotated images from previous step.

Step 3: determining a single (maximum) value of Sc for each rotated images.

Step 4: specifying a maximum value of Sc from all rotated images.

Step 5: determining the image with maximum Sc.

Step 6: extracting the angle that the image (I_{sc}) rotated according to it.

Step 7: determining the angle (θ) of the input image I_{in} .

4. RESULTS AND DISCUSSIONS

In this study, a webcam was employed. The hand is 90 cm away from the webcam. The static hand gesture dataset consisted of 569 images (361 for training and 208 for testing). Our approach implemented using the Computer Vision Toolbox in MATLAB. Six angles of hand rotation were identified ($0^\circ, 45^\circ, 90^\circ, 180^\circ, 270^\circ, \text{ and } 315^\circ$).

The Aggregate Channel Features (ACF) object detection approach is used to determine the hand's angle. The hand region is first manually extracted by the *Image Label Toolbox* in MATLAB as a first stage in angle detection, and then the computer is trained on the labeled images, resulting in a training file for the hand features. Then, based on the training file obtained in the training step, a simplified mathematical method utilized to detect the angles of hand gestures. Our approach recognizes the hand gestures and then calculates the angle of their direction. The input image (I_{in}) is rotated at various angles ($I_0, I_{45}, I_{90}, I_{180}, I_{270}, \text{ and } I_{315}$) during the test stage. The hand is then detected by comparing the rotated image with the produced file (.mat), which contains information about the hand's size and position. Then, depending on the number of rotation angles, (ACF) calculates the (Sc) values for each rotation direction, yielding a total of 6 (Sc) values for the input image. The maximum value represents the best match and identifies the hand angle of the input image. The degree of detection (Sc) on the Y-axis and the angle (θ) on the X-axis are graphed to show hand angle detection for a different input image.

Here, someone may wonder how to tell whether the original image was rotated at an angle is 0° or 180° . The proposed detection approach is dependent on the value of Sc, which is determining if the original image is at an angle 0° or 180° . As the Sc values for the original image (0°) are (70 at $\theta = 180^\circ$ and 109.5 at $\theta = 0^\circ$), whereas the resulted Sc values for the

original image (0°) are (65.5 at $\theta = 0^\circ$, and 94.38 at $\theta = 180^\circ$), the detection decision is for the maximum value as indicated by the dashed red line in Figure 7.

As shown in Figure 8, some results of detection appear with single peak as (315°) and others with double peaks as ($0^\circ, 45^\circ, 90^\circ, 180^\circ, \text{ and } 270^\circ$). The first sort, it is an inevitable result (single detection score). While the other type, which has two peaks, the decision of the detection notification is depends on the greatest peak, and thus the direction of the hand's direction is identified. On the Y-axis, the largest values of detection score (Sc) were 109.5, 109.5, 109.5, 94.38, 109.5, and 109.5, corresponding to angles of $0^\circ, 45^\circ, 90^\circ, 180^\circ, 270^\circ, \text{ and } 315^\circ$ respectively. Thus, for the rest of the results. The average detection time for each hand gesture angle and detection accuracy are shown in Table 1. It can be seen that angle (45°) required the shortest detection time (26.297 seconds for 29 images), whereas angle (180°) required the longest (31.318 seconds for 32 images).

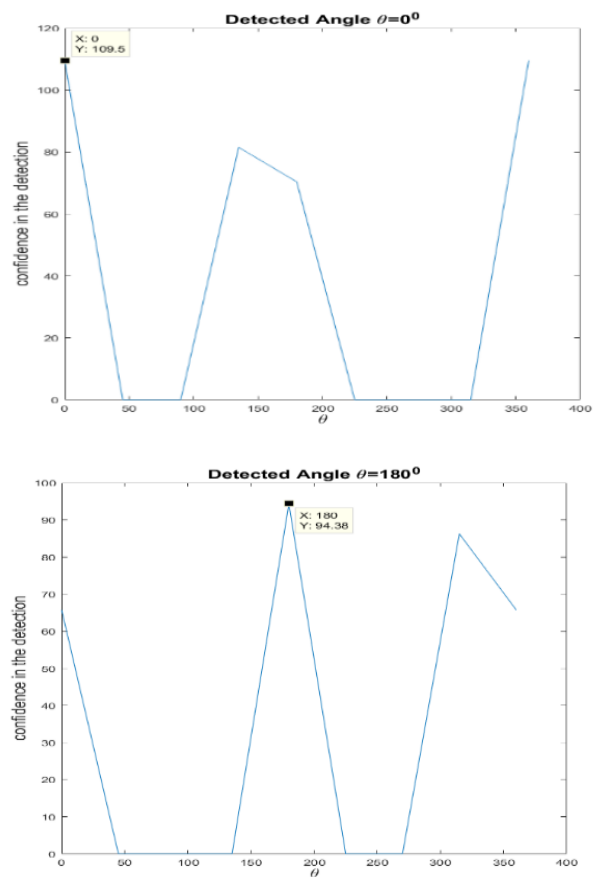


Figure 7. Decision of hand angle detection (0° and 180°)

Table 1. The average time cost to recognize the angles of hand gestures

| Hand gesture angle (Degree) | No. of images tested | Average detection time (Sec) | Detection accuracy (%) |
|-----------------------------|----------------------|------------------------------|------------------------|
| 0 | 34 | 0.9506 | 100 |
| 45 | 29 | 0.9068 | 100 |
| 90 | 35 | 0.9636 | 100 |
| 180 | 32 | 0.9787 | 100 |
| 270 | 38 | 0.9193 | 100 |
| 315 | 40 | 0.9482 | 100 |

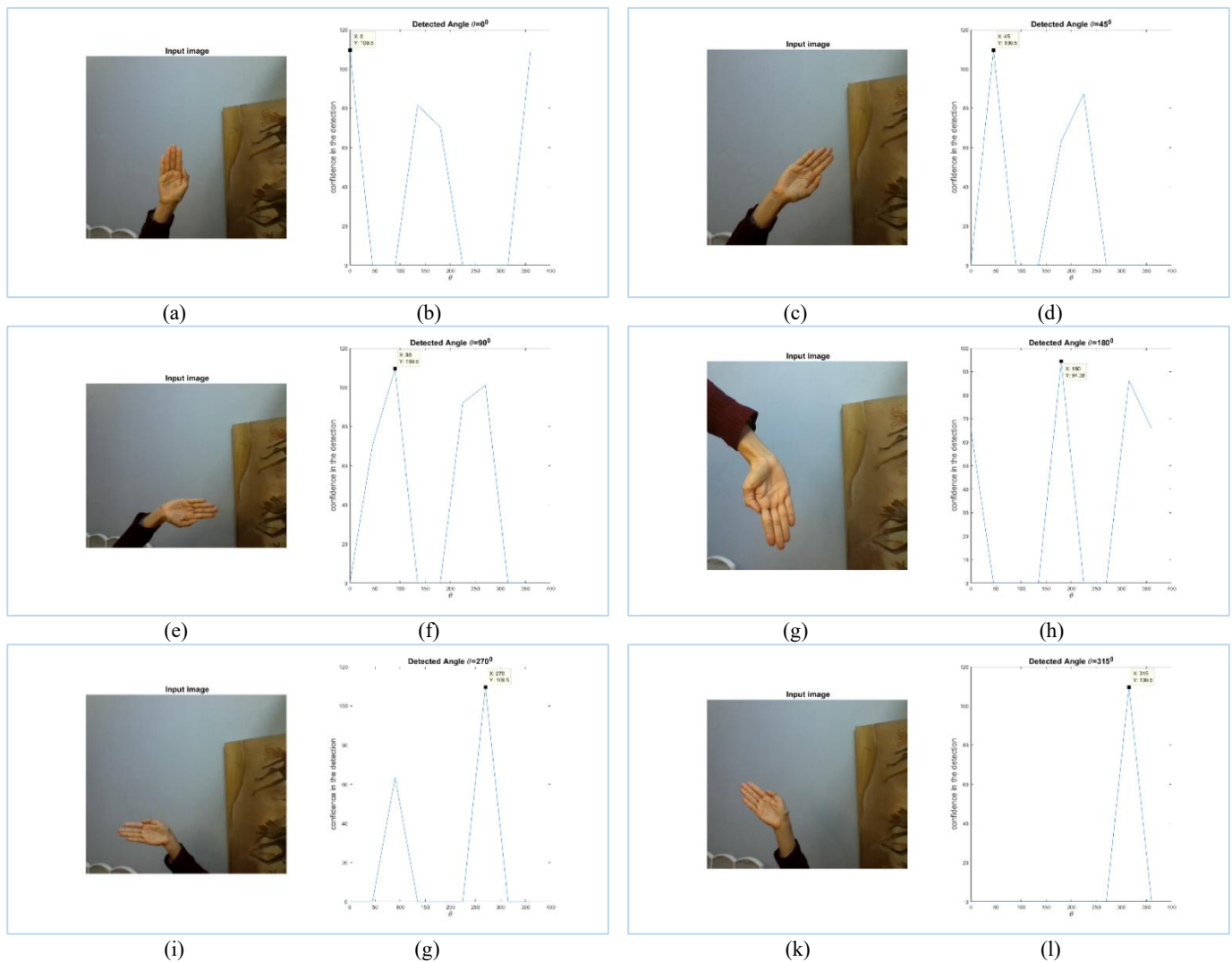


Figure 8. Determining the angle of hand gestures

5. CONCLUSIONS

In this paper, a new method is proposed to recognize the orientation of the angle of hand gestures, which can contribute in a variety of human-computer interaction applications. This study presents a vision-based hand gesture angles detection approach for recognition direction. A machine learning-based ACF detector has been proposed for this purpose. There are 208 color images of gestures (34, 29, 35, 32, 38, 40) were tested to identify hand gesture direction at angles of (0° , 45° , 90° , 180° , 270° , and 315°), respectively. Image labelling, computer training on a collection of 361 images, and ACF detector to determine the rotating hand angle are the three stages of our methodology. The method for detecting hand angle yielded considerable and extremely accurate results. The average detection time for all angles was 0.9445 seconds, and all hand angles were recognized with 100% accuracy.

According to the numerical results obtained, the numerical discoveries of our method excite and support a number of applications, including home automation, interactive virtual games, sign language, and robotic control.

ACKNOWLEDGMENT

The authors would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq) Baghdad-Iraq for its support in the present work.

REFERENCES

- [1] Wu, J., Qiao, G., Zhang, J., Zhang, Y., Song, G. (2013). Hand motion-based remote control interface with vibrotactile feedback for home robots. *International Journal of Advanced Robotic Systems*, 10(6): 270. <https://doi.org/10.5772/56617>
- [2] Erden, F., Cetin, A.E. (2014). Hand gesture based remote control system using infrared sensors and a camera. *IEEE Transactions on Consumer Electronics*, 60(4): 675-680. <https://doi.org/10.1109/TCE.2014.7027342>
- [3] Byun, S.W., Lee, S.P. (2019). Implementation of hand gesture recognition device applicable to smart watch based on flexible epidermal tactile sensor array. *Micromachines*, 10(10): 692. <https://doi.org/10.3390/mi10100692>
- [4] Wachs, J.P., Kölsch, M., Stern, H., Edan, Y. (2011). Vision-based hand-gesture applications. *Communications of the ACM*, 54(2): 60-71. <https://doi.org/10.1145/1897816.1897838>
- [5] Nagi, J., Ducatelle, F., Di Caro, G.A., Cireşan, D., Meier, U., Giusti, A., Gambardella, L.M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 342-347. <https://doi.org/10.1109/ICSIPA.2011.6144164>

- [6] Peng, Y., Tao, H., Li, W., Yuan, H., Li, T. (2020). Dynamic gesture recognition based on feature fusion network and variant ConvLSTM. *IET Image Processing*, 14(11): 2480-2486. <https://doi.org/10.1049/iet-ipr.2019.1248>
- [7] Zhang, Y., Zhou, W., Wang, Y., Xu, L. (2020). A real-time recognition method of static gesture based on DSSD. *Multimedia Tools and Applications*, 79(25): 17445-17461. <https://doi.org/10.1007/s11042-020-08725-9>
- [8] Taban, D.A., Al-Zuky, A., Kafi, S.H., Al-Saleh, A.H., Mohamad, H.J. (2021). Smart Electronic Switching (ON/OFF) system based on real-time detection of hand location in the video frames. In *Journal of Physics: Conference Series*, 1963(1): 012002. <https://doi.org/10.1088/1742-6596/1963/1/012002>
- [9] Oudah, M., Al-Naji, A., Chahl, J. (2020). Hand gesture recognition based on computer vision: A review of techniques. *Journal of Imaging*, 6(8): 73. <https://doi.org/10.3390/jimaging6080073>
- [10] Dawod, A.Y., Abdullah, J., Alam, M.J. (2010). Hand feature detection from skin color model with complex background. In *International Conference on Advances in Distributed and Parallel Computing Adpc*. <https://doi.org/10.5176/978-981-08-7656-2ATAI2010-29>
- [11] Viola, P., Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001*, 1, I-I. <https://doi.org/10.1109/CVPR.2001.990517>
- [12] Dollár, P., Appel, R., Belongie, S., Perona, P. (2014). Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8): 1532-1545. <https://doi.org/10.1109/TPAMI.2014.2300479>
- [13] Yutong, S., Zhao, H. (2015). Stock selection model based on advanced AdaBoost algorithm. In *2015 7th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 1-7. <https://doi.org/10.1109/ICMIC.2015.7409386>
- [14] Wu, P., Zhao, H. (2011). Some analysis and research of the AdaBoost algorithm. In *International Conference on Intelligent Computing and Information Science*, pp. 1-5. https://doi.org/10.1007/978-3-642-18129-0_1
- [15] Sager, C., Janiesch, C., Zschech, P. (2021). A survey of image labelling for computer vision applications. *Journal of Business Analytics*, 4(2): 91-110. <https://doi.org/10.1080/2573234X.2021.1908861>
- [16] Ribeiro, D., Nascimento, J.C., Bernardino, A., Carneiro, G. (2017). Improving the performance of pedestrian detectors using convolutional learning. *Pattern Recognition*, 61: 641-649. <http://dx.doi.org/10.1016/j.patcog.2016.05.027>
- [17] Viola, P., Jones, M.J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2): 137-154. <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>
- [18] Sharma, S., Singh, S. (2021). Vision-based hand gesture recognition using deep learning for the interpretation of sign language. *Expert Systems with Applications*, 182: 115657. <https://doi.org/10.1016/j.eswa.2021.115657>
- [19] Rehman, M.U., Ahmed, F., Khan, M.A., Tariq, U., Alfouzan, F.A., Alzahrani, N.M., Ahmad, J. (2021). Dynamic hand gesture recognition using 3D-CNN and LSTM networks. *Computers, Materials and Continua*, 70(3): 4675-4690. <https://doi.org/10.32604/cmc.2021.019586>
- [20] Zhang, Y., Zhou, W., Wang, Y., Xu, L. (2020). A real-time recognition method of static gesture based on DSSD. *Multimedia Tools and Applications*, 79(25): 17445-17461. <https://doi.org/10.1007/s11042-020-08725-9>
- [21] Verma, A., Hebbalaguppe, R., Vig, L., Kumar, S., Hassan, E. (2015). Pedestrian detection via mixture of CNN experts and thresholded aggregated channel features. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 163-171. <https://doi.org/10.1109/ICCVW.2015.78>
- [22] Hermawati, F.A., Tjandrasa, H., Suciati, N. (2018). Combination of Aggregated Channel Features (ACF) detector and faster R-CNN to improve object detection performance in fetal ultrasound images. *Int. J. Intell. Eng. Syst.*, 11(6): 1-10. <https://doi.org/10.22266/ijies2018.1231.07>
- [23] Saboo, S., Singha, J., Laskar, R.H. (2021). Dynamic hand gesture recognition using combination of two-level tracker and trajectory-guided features. *Multimedia Systems*, 28: 1-12. <https://doi.org/10.1007/s00530-021-00811-8>
- [24] Lazarou, M., Li, B., Stathaki, T. (2021). A novel shape matching descriptor for real-time static hand gesture recognition. *Computer Vision and Image Understanding*, 210: 103241. <https://doi.org/10.1016/j.cviu.2021.103241>
- [25] Mariappan, H.M., Gomathi, V. (2019). Real-time recognition of Indian sign language. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, 1-6. <https://doi.org/10.1109/ICCIDS.2019.8862125>
- [26] Athira, P.K., Sruthi, C.J., Lijiya, A. (2019). A signer independent sign language recognition with co-articulation elimination from live videos: An Indian scenario. *Journal of King Saud University-Computer and Information Sciences*, 34(3): 771-781. <https://doi.org/10.1016/j.jksuci.2019.05.002>
- [27] Tarchoun, B., Khalifa, A.B., Dhifallah, S., Jegham, I., Mahjoub, M.A. (2020). Hand-crafted features vs deep learning for pedestrian detection in moving camera. *Traitement du Signal*, 37(2): 209-216. <https://doi.org/10.18280/ts.370206>
- [28] Jonietz, C., Monari, E., Qu, C. (2015). Towards touchless palm and finger detection for fingerprint extraction with mobile devices. In *2015 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1-8. <https://doi.org/10.1109/BIOSIG.2015.7314623>
- [29] Ghorban, F., Marin, J., Su, Y., Colombo, A., Kummert, A. (2018). Aggregated channels network for real-time pedestrian detection. In *Tenth International Conference on Machine Vision (ICMV 2017)*, 10696: 106960I. <https://doi.org/10.1117/12.2309864>
- [30] Admasu, Y.F., Raimond, K. (2010). Ethiopian sign language recognition using artificial neural network. *2010 10th International Conference on Intelligent Systems Design and Applications*, pp. 995-1000. <https://doi.org/10.1109/ISDA.2010.5687057>
- [31] de Smedt, Q. (2017). Dynamic hand gesture recognition-From traditional handcrafted to recent deep learning approaches (Doctoral dissertation, Université de Lille 1, Sciences et Technologies; CRISAL UMR 9189). <https://hal.archives-ouvertes.fr/tel-01691715>.