ISSN 2303-4521
Periodicals of Engineering and Natural Sciences *Original Research*
Vol. 10, No. 32, April 2022, pp.467-476

Detection of hand gestures with human computer recognition by using support vector machine

Sura Abdulmunem Mohammed Al-Juboori[1], Hissah Almutairi[2,] Rasha Almajed[3], Amer Ibrahim[4], Hassan Muwafaq Gheni [5]
[1]Ministry Of Higher Education and Scientific Research, Baghdad, Iraq
[2] Department of IS, Jazan University, Jazan, Saudi Arabia
[3,4] College of Computer Information Technology; American University in Emirates, Dubai, UAE
[5] Computer Techniques Engineering Department, Al-Mustaqbal University college, Hillah 51001, Iraq

ABSTRACT

Many applications, such as interactive data analysis and sign detection, can benefit from hand gesture recognition. We offer a low-cost approach based on human-computer interaction for predicting hand movements in real time. Our technique involves using a color glove to train a random forest classifier and then predicting a naked hand at the pixel level. Our algorithm anticipates all pixels at a rate of around 3 frames per second and is unaffected by differences in the surroundings. It's also been proven that HCI-based data augmentation is more effective than any other way for enhancing interactive data. In addition, the augmentation experiment was carried out on multiple subsets of the original hand skeleton sequence dataset, each with a different number of classes, as well as on the entire dataset. On practically all subsets, the proposed base architecture improved classification accuracy. When the entire dataset was used, there was even a modest improvement. Correct identification could be regarded as a quality indicator. The best accuracy score was 94.02 percent for the HCI-model with support vector machine (SVM) classifier.

Keywords: Sign detection, Hand gestures, Support vector machine, Human computer interaction

Corresponding Author:

Hissah Almutairi
Department of IS
Jazan University
Jazan, Saudi Arabia
E-mail: halmutairi@jazanu.edu.sa , hesshalmutairi@gmail.com

1. Introduction

Hand detection and recognition, a crucial feature of vision-based gesture recognition for natural Human-Computer Interaction, is conceivable (HCI). It's worth emphasizing that the hands are necessary organs for humans to interact and carry out several important tasks daily [1]. Hand and other body part tracking from digital photographs has transformed not only the scientific world, but also the entertainment business, such as gaming and animation. It also replaced traditional hard labor in fields like as factory automation, virtual reality, rehabilitation and handicap support, performance monitoring, and many others [2]. There are now methods in which a user can have unique hardware connected to his or her body that allows for accurate study of the joints and geometry of body components. It's neither natural nor comfortable, though. Furthermore, wearing specialized hardware tailored for specific tasks is very costly. For simplicity, cost effectiveness, and a natural interface experience, vision-based applications are favored over alternative approaches. In [3], the authors use sophisticated computer graphics to generate training samples. We found this expensive and using color gloves, Developers can make their own custom gestures with ease. Another advantage of our approach is that the system uses actual raw depth images as training samples. It takes up realistic noises like shadows and instrument noise

on its own. It is extremely difficult to mimic these noise effects using computer graphics, as shown in [3]. Our contributions are summarized as follows:

1. From a computational standpoint, there appears to be no literature comparing SVM to random forest. We present an in-depth complexity analysis of the two techniques, rather than simply providing experimental accuracy as is done in typical machine learning studies.
2. By carefully exploring a broad space of parameters, we undertake lengthy trials to evaluate the effectiveness of the random forest classifier. The results are intriguing, as they lead to a better understanding of the support vector machine.

2. Background

With complex background, one can observe skin model and additionally use assumption that the ROI is always in motion as described in [4]. There are cases where false skin detection under different lighting conditions in uncontrolled situations can occur. This can be identified with motion segmentation with finding difference between moving foreground and stationary background [5]. This is brilliant idea for hand segmentation to reduce false skin detection and can certainly implementable. As suggested by [6, 7] two different approaches to track motion pixels; the first one is simple frame subtraction method. Although it has limitation as it's sensitive to camera noise, additionally hand or face movement will open and occlude regions in background and can contribute in some sort of false detection [8]. The second method is more robust with optical flow motion tracking, where they need to initialize feature points. We have tried this approach with HCI with our case and experience suggests that it is good for hand motion where you can define multiple tracking points. But for hand, since it has smaller size compared to its background which include face, this method is not that much helpful. Also, it will place limitation of detection algorithm since ROI must be in motion. As described in their paper [9], presents very good example of skin probability function and its uses, and how it can track wrong regions in a scene. They mention about Hue color space where he is independent of intensity and can be ideally suitable for skin detection [10]. The problem arises when along with user there are object which has skin color tone. For example, a table may have certain wood like color which can be misidentified as skin color tone with Hue thresholding. Now if we are using skin probability function, where it represents the lighter pixels as skin and darker as background [11]. The hand region has darker pixels compare to the table in a background and if we solely rely on probability function where, higher probability will indicate the table region in background and that means a weak skin detector. That's why they introduce the motion detectors which can identify moving region of interest as compared to static background. As saw in Figure 1 Hand gesture recognition using learning mode and recognition algorithm.

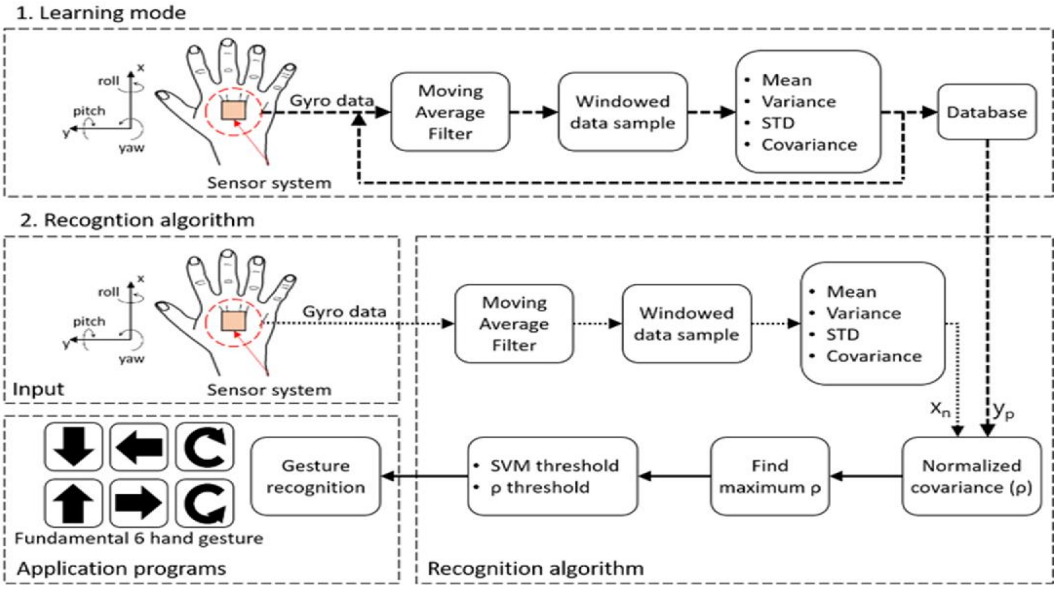


Figure 1. Hand gesture recognition using learning mode and recognition algorithm [11]

In [12] author calculates depth histogram and thresholds depth image based on that. They assume that the hands are in front of the camera all the time and the rest of human body is considered as part of background. They calculate distance of hands and other body parts from accumulated histogram by analyzing rapid increase in the histogram slops as described in [13]. They assume that the hands are in front of camera and if slope is greater than certain value then this region is selected as candidate ROI. They use color information to segment other objects at the same depth level, thus they turn to skin color detection. This method is highly restrictive for hand movement since it requires hands to be in front of camera all the time. It deals solely with depth images and finds hand and its movement as described in [14]. They propose classifier that combines a boosting and cascade structure. The features for training are depth differences at the stage of detection and learning. They also implement depth region grow and Depth Adaptive Mean Shift. Although they claim successful results the tracking algorithm can be quite slow because of region growing method.

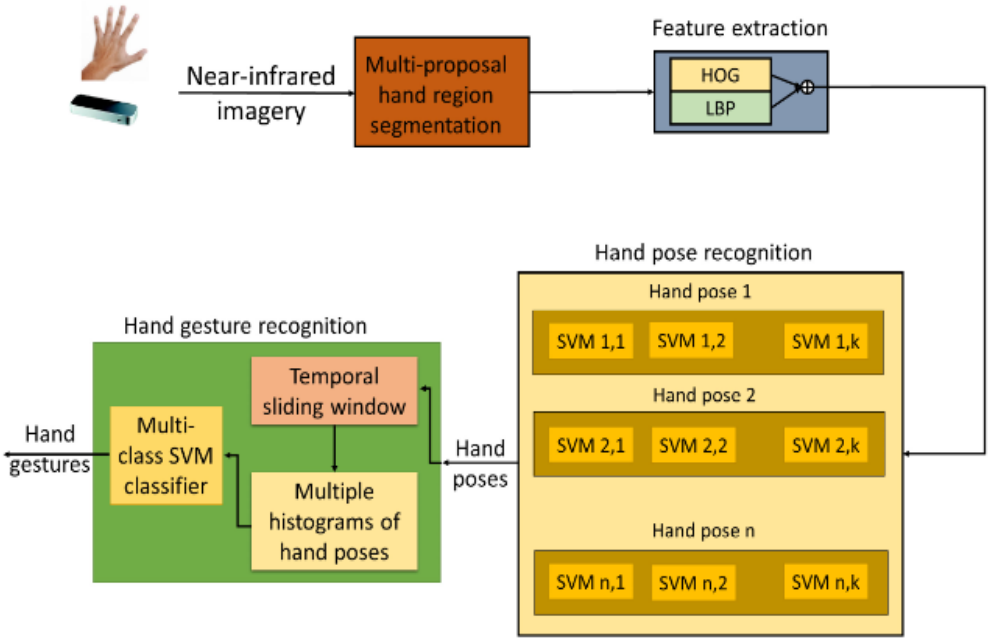


Figure 2. A complete block diagram of gesture understanding algorithm [15]

As described in their paper [16, 17], presents very good example of skin probability function and its uses, and how it can track wrong regions in a scene. They mention about Hue color space where he is independent of intensity and can be ideally suitable for skin detection [17]. The problem arises when along with user there are object which has skin color tone. For example, a table may have certain wood like color which can be misidentified as skin color tone with Hue thresholding. Now if we are using skin probability function, where it represents the lighter pixels as skin and darker as background [18]. The hand region has darker pixels compare to the table in a background and if we solely rely on probability function where, higher probability will indicate the table region in background and that means a weak skin detector. That's why they introduce the motion detectors which can identify moving region of interest as compared to static background.

In [19] author calculates depth histogram and thresholds depth image based on that. They assume that the hands are in front of the camera all the time and the rest of human body is considered as part of background. They calculate distance of hands and other body parts from accumulated histogram by analyzing rapid increase in the histogram slops as described in [20]. They assume that the hands are in front of camera and if slope is greater than certain value then this region is selected as candidate ROI. They use color information to segment other objects at the same depth level, thus they turn to skin color detection. This method is highly restrictive for hand movement since it requires hands to be in front of camera all the time [21-24].

3. Methodology

In this paper, understanding gestures is a key challenge with ML (machine learning) and HCI in this study (human-computer interaction). Gestures are used by both humans and animals to show emotions and, in some situations, to communicate. Humans engage with the natural world by using their hands as tools. The human hand, as we all know, is much smaller than the rest of the body and has the most degrees of freedom. As a result, modeling the human hand would be difficult, even if most real-world applications do not necessitate many degrees of freedom., In our case of the hand, for example, elaborate manual modeling is not required in this regard. A gesture might be static, dynamic, or a combination of the three. Hand movements will not be required for stationary gestures, but rather a complicated set of specific locations / signals expressing various symbols or meanings, as indicated in the diagram. Others, on the other hand, include hand movements up to a specific point to establish a trajectory. In our instance, we must recognize that human gestures must coexist with those working on manufacturing lines. As a result, in this section, we'll demonstrate how to decode pre-programmed simple signals using basic hand path movements.

- Once hand has been initialized, hand tracking algorithm will return 3D points.
- Store every 15 points and find out if the start and stop point are still in the range along with others, if so, then raise the flag for gesture to start.
- Once flag has been raised for start process, record all consecutive points, and inform user with text on screen and draw trajectory with different(red) color.
- Again, store every 15 points while trajectory is being recorded, and check continuously if the hand has stopped moving.
- Once that condition realizes raise other flag for gesture-stop and reset flag for gesture-start making way for new gesture.

3.1. Dataset description

IEEE-Data Port, an open-source archive for hand recognition, provided the data. We were able to run experiments with similar (scalability) results after processing this dataset. The total number of scales was divided in half, which was offered as a heuristic solution. The first half, which contains photos of large sizes, would be parallelized by computing each scale in parallel, while the second half, which contains images of small scales, would also be parallelized. The dataset was processed in this way using multi-cross folds.

3.2. Model

For this, we employed a support vector machine (SVM) classifier. SVM was chosen due to its ease of use, predictability, and speed; also, HCI has an excellent interface for creating classifiers. It will be examined in many ways once a complete set of 3D data has been recorded as a single trajectory. Individual points are scanned and used, then acceptable angles between successive points are found, as explained in the Data collection and Processing sections. Later, for better representation, those angles are translated to codes. As a result, we will obtain a sequence of codes that will constitute the feature vector for both training and testing our SVM from a set of points. The entire procedure is described in detail.as we saw in Figure 2 the hand gestures recognition model.

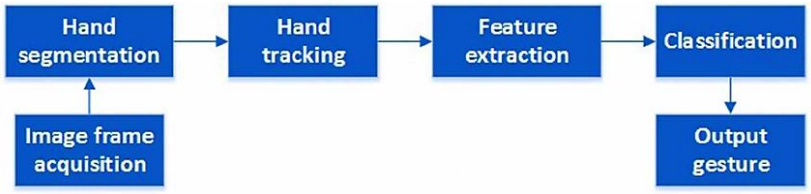


Figure 3. A systematic and pictorial view of hand gestures recognition model

3.3. Pruning and overfitting

The impact of trimming on prediction accuracy was investigated. We pruned the forests trained with 1000 features and three trees for varied numbers of training images to investigate this. The pruned models' test set accuracy was then calculated. When compared to the whole tree, trimming the tree revealed absolutely no difference in accuracy. The reason for this, we believe, is because the splitting nodes very high up in the tree have already confidently divided the training points into their class. The random forest method, on the other hand, continues to exhaust the features until the full tree is built. We don't lose accuracy by halting the prediction early in the tree, and we enhance prediction speed. Finally, we calculated the training accuracy and compared it to the test accuracy to see how much the model overfits to the training data. The number of training samples was changed while 1000 features and three trees were fixed. We illustrate that there is a lot of overfittings going on. The training error is continuously low (mean 2.03%) and far less than the testing accuracy.

Tables 1. calculate the Time of performance

Process	**Time**
Gesture Recognition	0.07
Detection	7.3
Classification	15.3
Total	22.67

4. Results

When trained on the expanded training sets, the SVM-based recognition classifier leveraging HCI design parameters enhanced its average maximum validation accuracy. The model with the best generator loss attained an average maximum validation accuracy of 94.02 percent. As a result, the least generator loss model outperformed the model with the best inception accuracy, giving the impression that it is better suited for selecting the optimal HCI model.

Without data augmentation, the multi-class HCI findings represent a considerable improvement over the original baseline classifier and warrant further exploration.so as we saw detect and recognize the hand from figures so each one has detected different for each other in figure 4 we see no hand detect and in Figure 5 we see Thumbs Down, and in Figure 6 we see Back Peace Sign.

Table 2. Classification accuracies on original test set for different augmentation methods with a multi-class SVM outperform

Class.	Augmentation	Model	Ratio		Best Val%	SD Mean	Last Val%	SD Mean
			Orig.	Gen.				
20-24	-	-	1	0	83.22	1.22	**83.80**	3.50
20-24	SVM	HCI	1	1	**94.02**	2.23	80.05	5.08
20-24	Cut Length	HCI	1	1	83.81	1.22	79.40	2.00
20-24	Add Noise	HCI	1	1	83.16	4.77	79.00	2.99
20-24	Add Noise	HCI	1	1	82.84	3.87	81.00	4.99
20-24	Add Noise	HCI	1	1	82.84	2.71	81.22	4.00
20-24	Random Forest	HCI	1	1	83.16	1.65	81.31	5.00

Table 2 shows the results for combining HCI augmentation methods. The tests were conducted on the classes 20 to 24 as the alternative augmentation methods were effective for this subset in the previous experiments. For each augmentation method, 60 new samples were independently generated and appended to the real dataset. The improvement that was achieved with the multi-class SVM was also observable when it was used together with the alternatively generated sequences. The combination of sampling from random sequence segments and SVM augmentation was even slightly more effective than the multi-class SVM on its own. Interestingly, there was a strong boost in the best validation accuracy when all three alternative methods were applied together. The

resulting score is in the same region as for the multi-class SVM. Potentially, the classifier was able to learn new information from the three combined transformations that can all be concurrently exercised by the multi-class SVM. Finally, using all four augmentation methods did not lead to a further boost. Overall, it appears that the multi-class data augmentation comprises most of the positive augmentation effects by itself.

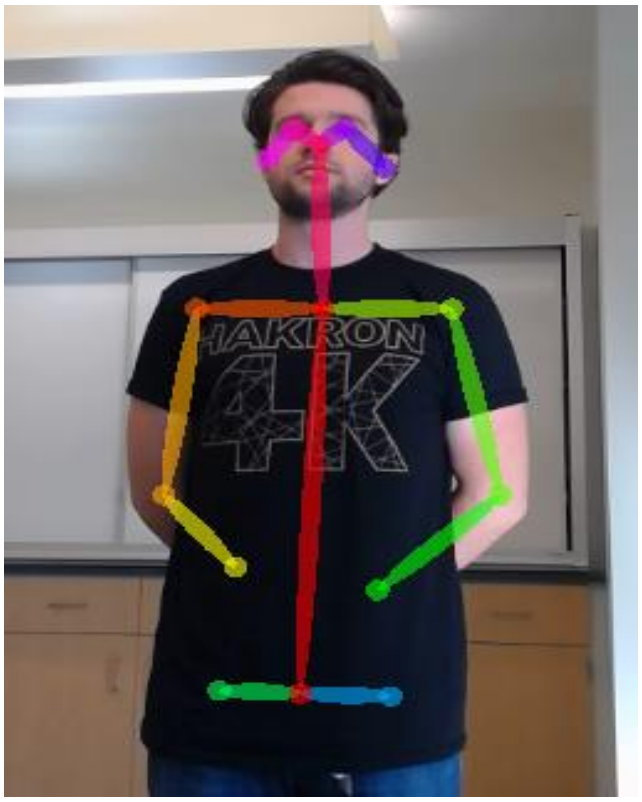


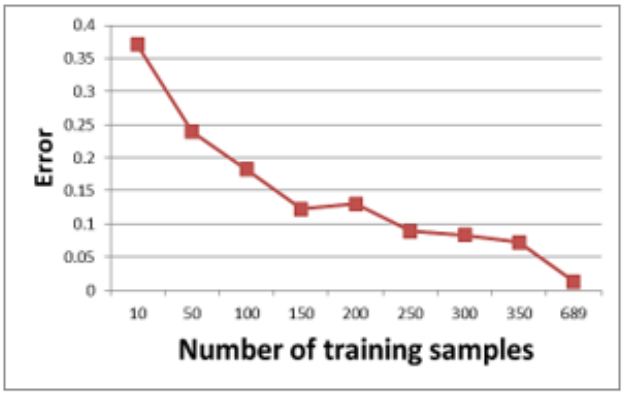
Figure 4. From an image, we see no hand detect

Figure 5. From an image, we see thumbs down

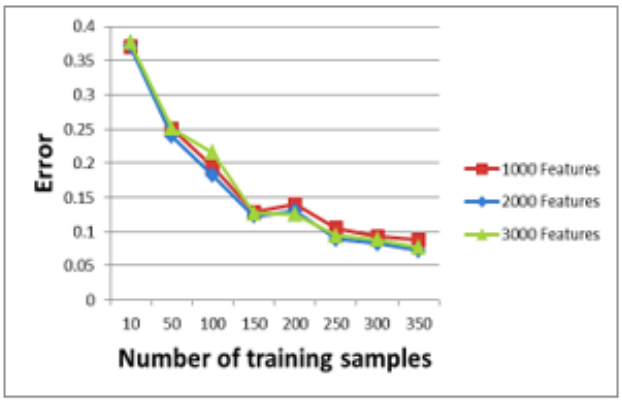
Figure 6. From an image, we see back peace sign

The two statistics appear to connect with absolute improvement when focused on the average of the best validation error. On the one hand, it appears that as the number of classes and hence the number of raw samples in the training set grows, the improvement decreases. The difference grows less and even negative when the number of selected categories increases from seven to 10. Using data augmentation for twenty classes as well as the complete data set improves classification accuracy, even though the categorization is quite modest.
Data augmentation, on the other hand, is more effective for subgroups with longer maximum sequence lengths. The maximum sequence length differed dramatically between the three subgroups of the five classes, even though they all had the same amount of original training samples. The most significant improvements were made in the classes with the longest maximum sequence length. While the maximum sequence length was the same in the 7th, 8th, 9th, and 10th grade trials, where improvement decreased as the training sample size grew, when the trials were repeated across grades, the maximum sequence length was much longer. Increasing the data again has a good effect in these circumstances. Synthetic samples are less effective for improving classification accuracy for the total data set because the maximum number of frames for created sequences is cut to a smaller number. When synthetic sequences contain a maximum of 500, 700, or 1000 frames, all classification scores are lower than the original baseline.

These broad observations make it difficult to draw definite conclusions, and it appears that the effectiveness of data augmentation is likely to be influenced by a variety of other factors, most notably the shape, kind, and diversity of the classes chosen. As a result, determining a definite limit beyond which the process of increase begins to fail is difficult. Data augmentation, on the other hand, is clearly more useful for tiny data sets that are easy to supplement with additional information. While the lower training set reduces the visual quality of the synthetic samples, the generated sequences appear to carry enough information to have a favorable effect.



(a) Varying number of training samples.



(b) Varying the number of features. Number of features: 3000, and number of trees 3.

Figure 7. Experimental results being displayed in graphical representation

As we see in Figure 7 the Number of training samples explain the effect of varying the number of training samples and Number of features explain the effect of varying the number features.

Table 3. comparison our work with provided papers

Article	Technique	Accuracy
[21]	Deep conventional neural network (DCNN)	93.25%
[22]	LSTM	99.4%
Proposed	**Support vector machine (SVM)**	**94.02%**

5. Conclusion

In this research paper, we constructed a human-computer interaction system for hand gesture detection, and we tested augmentation strategies on a subset of three classes. Because the dataset was smaller, training time was lowered, allowing for more trials to be completed in the allotted time. It was discovered that training all three classes on a single multi-class HCI is more successful for augmenting data than using a separate network to generate samples for each class. Because multi-class HCI uses a larger number of training samples and different classes, it is more likely to capture both the underlying structure of a hand position and the unique variations across the action classes. Furthermore, even though its synthetic sequences appeared less realistic, the HCI model with the lowest generator loss had a larger beneficial influence on classification accuracy (94.02%). Overall, the thesis's principal goals were accomplished. The established architecture was put into place and proved to be beneficial in terms of data augmentation. Furthermore, the HCI was evaluated and compared to a variety of alternatives using appropriate assessment measures.

Declaration of competing interest

The authors declare that they have no any known financial or non-financial competing interests in any material discussed in this paper.

References

[1] D.-S. Tran, N.-H. Ho, H.-J. Yang, S.-H. Kim, and G. S. Lee, "Real-time virtual mouse system using RGB-D images and fingertip detection," Multimedia Tools and Applications, vol. 80, no. 7, pp. 10473–10490, Nov. 2020.

[2] J. Schulte, M. Kocherovsky, N. Paul, M. Pleune, and C.-J. Chung, "Autonomous Human-Vehicle Leader-Follower Control Using Deep-Learning-Driven Gesture Recognition," Vehicles, vol. 4, no. 1, pp. 243–258, Mar. 2022.

[3] G. Benitez-Garcia, L. Prudente-Tixteco, L. C. Castro-Madrid, R. Toscano-Medina, J. Olivares-Mercado, G. Sanchez-Perez, and L. J. G. Villalba, "Improving Real-Time Hand Gesture Recognition with Semantic Segmentation," Sensors, vol. 21, no. 2, p. 356, Jan. 2021.

[4] K. Yang, M. Xu, X. Yang, R. Yang, and Y. Chen, "A Novel EMG-Based Hand Gesture Recognition Framework Based on Multivariate Variational Mode Decomposition," Sensors, vol. 21, no. 21, p. 7002, Oct. 2021.

[5] Y. Dong, J. Liu, and W. Yan, "Dynamic Hand Gesture Recognition Based on Signals From Specialized Data Glove and Deep Learning Algorithms," IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1–14, 2021.

[6] K. Zhang and F. Chen, "Research on sEMG Gesture Recognition Based on Hybrid Dilated Convolutional Neural Network Combining Bidirectional Gated Recurrent Unit And Attention Mechanism," 2021 China Automation Congress (CAC), Oct. 2021.

[7] D. Bhavana, K. Kishore Kumar, M. Bipin Chandra, P. V. Sai Krishna Bhargav, D. Joy Sanjana, and G. Mohan Gopi, "Hand Sign Recognition using CNN," International Journal of Performability Engineering, vol. 17, no. 3, p. 314, 2021.

[8] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed, R. Damaševičius, R. Maskeliūnas, and K. H. Abdulkareem, "Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model," Applied Sciences, vol. 11, no. 9, p. 4164, May 2021.

[9] M. Murugeswari and S. Veluchamy, "Hand gesture recognition system for real-time application," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, pp. 1220-1225 , May 2014.

[10] J. Li et al., "Internet of things assisted condition-based support for smart manufacturing industry using learning technique," Comput. Intell., vol. 36, no. 4, pp. 1737–1754, 2020, doi: 10.1111/coin.12319.

[11] I. Al_barazanchi, Z. A. Jaaz, H. H. Abbas, and H. R. Abdulshaheed, "Practical application of iot and its implications on the existing software," Int. Conf. Electr. Eng. Comput. Sci. Informatics, vol. 2020-Octob, no. October, pp. 10–14, 2020, doi: 10.23919/EECSI50503.2020.9251302.

[12] H. D. Nguyen, Y. C. Kim, S. H. Kim, and I. S. Na, "A method for fingertips detection using RGB-D image and convolution neural network," 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Jul. 2017.

[13] P. Sun, W. Zhang, H. Wang, S. Li, and X. Li, "Deep RGB-D Saliency Detection with Depth-Sensitive Attention and Automatic Multi-Modal Fusion," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2021.

[14] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, and M. S. Hossain, "Hand Gesture Recognition Using 3D-CNN Model," IEEE Consumer Electronics Magazine, vol. 9, no. 1, pp. 95–101, Jan. 2020.

[15] T. Mantecón, C. R. del-Blanco, F. Jaureguizar, and N. García, "A real-time gesture recognition system using near-infrared imagery," PLOS ONE, vol. 14, no. 10, p. e0223320, Oct. 2019.

[16] H. Tao et al., "A Newly Developed Integrative Bio-Inspired Artificial Intelligence Model for Wind Speed Prediction," IEEE Access, vol. 8, pp. 83347–83358, 2020, doi: 10.1109/ACCESS.2020.2990439.

[17] P. Neto, D. Pereira, J. N. Pires, and A. P. Moreira, "Real-time and continuous hand gesture spotting: An approach based on artificial neural networks," 2013 IEEE International Conference on Robotics and Automation, May 2013.

[18] A. Banerjee, A. Ghosh, K. Bharadwaj, and H. Saikia, "Mouse Control using a Web Camera based on Colour Detection," International Journal of Computer Trends and Technology, vol. 9, no. 1, pp. 15–20, Mar. 2014.

[19] U. Beyaztas, S. Q. Salih, K. W. Chau, N. Al-Ansari, and Z. M. Yaseen, "Construction of functional data analysis modeling strategy for global solar radiation prediction: application of cross-station paradigm," Eng. Appl. Comput. Fluid Mech., vol. 13, no. 1, pp. 1165–1181, 2019, doi: 10.1080/19942060.2019.1676314.

[20] A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, Eds., "Consumer Depth Cameras for Computer Vision," Advances in Computer Vision and Pattern Recognition, 2013.

[21] M. M. Islam, M. R. Islam, and M. S. Islam, "An Efficient Human Computer Interaction through Hand Gesture Using Deep Convolutional Neural Network," SN Computer Science, vol. 1, no. 4, Jun. 2020.

[22] Z. Yang and X. Zheng, "Hand Gesture Recognition Based on Trajectories Features and Computation-Efficient Reused LSTM Network," IEEE Sensors Journal, vol. 21, no. 15, pp. 16945–16960, Aug. 2021.

[23] A. S. Abdullah, M. A. Abed, and I. Al Barazanchi, "Improving face recognition by elman neural network using curvelet transform and HSI color space," Period. Eng. Nat. Sci., vol. 7, no. 2, pp. 430–437, 2019.

[24] S. S. Oleiwi, G. N. Mohammed, and I. Al-barazanchi, "Mitigation of packet loss with end-to-end delay in wireless body area network applications," Int. J. Electr. Comput. Eng., vol. 12, no. 1, pp. 460–470, 2022, doi: 10.11591/ijece.v12i1.pp460-470.