Control of Computer Pointer Using Hand Gesture Recognition in Motion Pictures

Yalda Foroutan, Ahmad Kalhor, Saeid Mohammadi Nejati, Samad Sheikhaei

School of Electrical and Computer Engineering, University of Tehran, Iran

Abstract

A user interface is designed to control the computer pointer using hand detection and classification of its gesture. A dataset with 6720 image samples is collected, including four categories: fist, palm, left, and right. The images are captured from 15 individuals in simple backgrounds and different perspectives and light conditions. A CNN network is trained on the dataset to predict a label for each captured image and measure its similarity with other samples. Finally, commands are defined so that the user can click, right-click and move the cursor based on their hand gestures. The algorithm reaches 91.88% accuracy and can be used in different simple backgrounds.

Keywords: Hand Gesture Recognition, Dataset, Convolutional Neural Network, Human-Computer Interaction, Classification

1. Introduction

Due to technological advances, computers nowadays play a vital role in human life and penetrate in general aspects of their personal and social lives. The increasing availability of personal computers along with mass-produces have a growing impact on daily life, leading to a multidisciplinary field calling Human-Computer Interaction (HCI), which focuses on improvements of human and computer communications, such as creating an interface to transform hand gestures into meaningful commands. Despite HCI, intermediaries for computers such as keyboard and mouse have still been used. Although much significant progress has been made to improve the mouse when the first mouse was introduced, this hardware is based on direct contact and curbs users to control a computer from a close distance. Now, the COVID-19 virus

pandemic seems to accelerate the shift towards the new era of contact-based gadgets that control machines and computers.

2. Literature Review

In recent years, machine learning enthusiasts track human activities [1], some of these activities present alternative ways to control computers. For instance, they were using a Kinect sensor [2], EEG mouse [3], and EMG signals [4] to classify human actions and allocate mouse commands to them to change the position of the cursor. The methods mentioned above, however, remove the ordinary hardware but use other hardware, which is more expensive and bulkier than the usual mouse. Thus, it would be superior to develop software for replacing the mouse and controlling the computer pointer.

Since one of the most useable organs of the human body for doing daily tasks is their hands, people can also benefit from them to guide intelligence systems. Thus, one effective way to replace the hardware of the mouse could be hand gesture recognition applications to control the computers. Applications using hand gesture recognition follow two approaches: based on contact or machine learning methods without a single contact. Older works were following the former approach, such as a data glove [5], which detects hand gestures and tracks their movement. However, these gadgets have weaknesses, their price along with the creation of hand movement restrictions because of their weights, wiring, and sensors. Due to computer vision advances and obtaining more information from images, these kinds of gloves have become much simpler. Once the wiring was removed, a camera has been added in order to track hand movements [6]. In fact, an algorithm, taking advantage of a camera, can learn to chase the color of the glove or the shape of the palm and fingers. After that, even the glove was replaced with some colored fingertips [7]. The tips, eventually, were also removed, and hand gesture applications become touch-less and based on machine learning techniques by the use of a camera and its captured frame.

Therefore, machine learning techniques using image processing systems like cameras can be an alternative for contact-based approaches. The most valuable feature of these techniques could be degrees of freedom for hand movements, which should be obligatory for the HCI systems. Such techniques are combined with image processing and computer vision methods. Image processing methods convert the captured images of a camera to digital forms as well as implement scaling, filtering, and noise remover. Computer vision

can create eyesight for computers to distinguish between different gestures in the same way humans do. For instance, in [8], human skin is detected by setting a color threshold for the skin color, then the backgrounds of each frame are removed. In [9], there is no learning algorithm, and a video sequence, instead of frames, is imported to the processing section. Also, methods like skin detection and approximate median model are used. [10] detects both hand and head based on the skin color and creates a white and black mask from each frame. Next, a VGGNet model is trained to differentiate between hand and head. In [11], an angle between thumb and index finger is defined for discriminating hand gestures. In some other articles, such as [12], frames convert to the HSV color space to access each color only in one component. In [13], a blue background is used to assist an algorithm with the difference between human skin and the background in HSV space.

Hand gesture recognition usually has two stages, such as hand localization and gesture classification. SIFT is one of the fastest tools for feature extraction before the emergence of neural networks. In [14], the use of SIFT leads to distinguish between eight gestures to control home appliances like fans and washing machines. Also, in [15] optical flow is used to classify hand gestures, though for tracking hand movements, this approach may wrongly follow another object, like the user's head with which the hand has overlap. Although the color-based methods are simple and easy to implement, their robustness along with generalization from person to person or background to background could be low. These methods, for instance, have a skin-color dependency, like [16], which users should manually import their skin colors. Furthermore, pixel-wise differentiation between human skin and backgrounds is more sophisticated and sensitive than it seems.

The availability of massive datasets has increased the applications of neural networks and make them a substitute way for classic machine learning ones since they can be invariant to light conditions, perspective variations, and different backgrounds. Now, hand detection algorithms can be benefited from object detection algorithms that are based on neural networks. Algorithms such as You Only Look Once (YOLO) and Single Shot Multi-Box Detector (SSD) are working for real-time tasks and have higher accuracy as opposed to RCNN or Faster RCNN. In addition, the SSD algorithm can be learned by new images [17]. [18] applies two SSD detectors: The first one detects head and shoulder area, the second one recognizes hand gestures in the detected area. Even though the computational cost of these algorithms

may be high, techniques like selective-dropout can reduce the computational load for a decrease in performance lag [19]. If hand detection and gesture recognition be merged during the detection part, their output will be a valid predicted label [20]. Hand gesture recognition algorithms based on deep learning are more accurate than classic methods. Although deep learning approaches do not require feature extraction and reduce the requirements of handy designs, they can be slow because of the complexities. Thus, a data glove and hand gesture recognition by means of an SSD hand detector and SVM, respectively, are used to classify gestures of both hands [21].

3. Proposed Algorithm

In this section, a human-computer interface based on hand gestures for the computer's pointer is designed. A dataset is collected to train a convolutional neural network (CNN) for two purposes: a classifier and similarity calculator networks. Finally, a command controller is designed to convert a predicted label to a cursor task.

3.1. Dataset

A hand dataset with 6720 image samples (300 x 300) of 15 subjects in 18 different simple backgrounds is collected. The dataset has four different gesture classes, such as fist, palm, right and left. Subjects were asked to photograph from both hands and use both palmar and dorsal sides. Figure 1 depicts four gesture samples from the dataset. The dataset has 5120 training and 1600 validation samples (800 for validation, 800 for test). Although it is common to split training and validation sets randomly, in this project, they have different distribution to avoid overfitting and boost the generalization power since validation samples are more likely to in real-time frames. In fact, the training samples are captured from webcams, without any intermediary software, and the validation samples are the acceptable frames from the SSD hand detector. Therefore, the CNN classifier is learned by a distribution and copes with another one.

3.2. Hand Detection

When a user moves their hand in front of a computer webcam, the algorithm captures frames, which are then preprocessed and imported to an SSD hand detector. If a hand is detected in a frame, two outputs would be expected from SSD, such as a cropped frame and the center coordinate of the

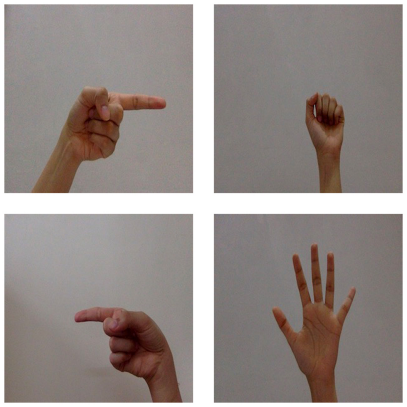

Figure 1: Examples from the proposed hand dataset-Samples are captured from different cases with simple backgrounds in various light conditions and distances from webcams.

cropped frame. Since the detector plots a bounding box around a hand in each frame, the hand would be cropped from the bounding box edges. The cropped frame is then fed to a classifier, and the center coordinate would be used in the command controller to move the computer cursor. Unless a hand is detected in a frame, the next frame will be considered for detecting a hand. The process of hand detection is shown in Figure 2.

3.3. Classification

Like any other image classification task, this project should deal with how to classify a sample to one of the defined classes or ignore that if it belongs to an undefined one. In other words, classifiers are trained based on a close dataset with limited numbers of samples and classes instead of having all the possible ones. Here, because the SSD considers all kinds of hand gestures, the algorithm faces two types of cropped frames imported to its classification part: one of four defined classes in the dataset or undefined ones. In fact, four defined classes must become pointer commands, and no action must occur for other given gestures. Thus, a CNN is trained to predict a valid label for frames with defined classes and remove others. The architecture of EfficientNet-B0 is used, followed by 8 fully-connected layers or classification [22]. The last layer of EfficientNet-B0 has 1280 neurons, which are reduced to 4 neurons by 8 fully-connected layers as opposed to 2 layers for reaching higher generalization power [23]. The training samples with 70 x 70 x 3 feature maps train the CNN network in 20 epochs, and the CNN reaches

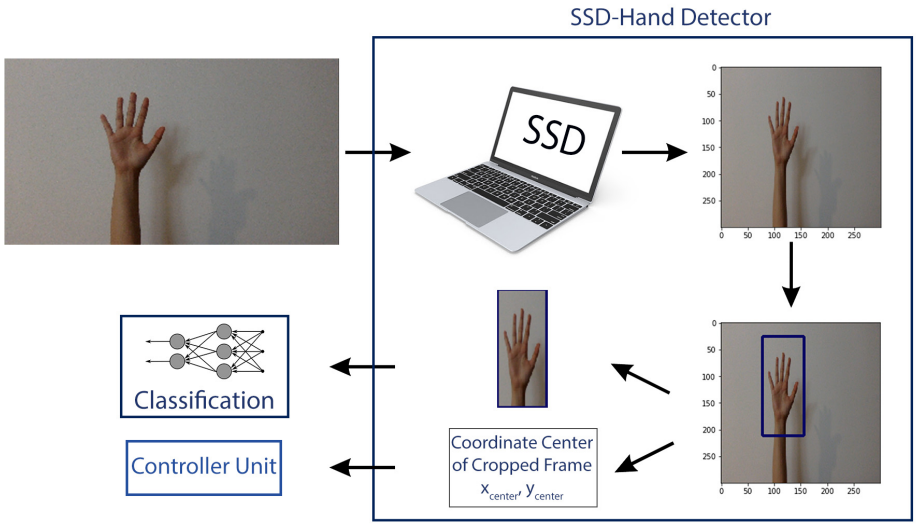

Figure 2: The taking steps of the SSD hand detector-A captured frame is fed through the SSD hand detector, and two outputs are extracted from that, such as a cropped frame and center coordinate of the frame.

99% accuracy for the test set. Therefore, captured frames are preprocessed, and if they pass the SSD, cropped version of the will feed to the frozen CNN network for being classified.

A Radial Basis Function (RBF) network is designed to eliminate the undesired frames. After the classifier is trained and frozen, its last 8 dense layers are removed to form a similarity network. In fact, a model is required to act as an encoder or a feature extractor, which diminishes the dimensions of the samples from 70 x 70 x 3 to 1280. Then, samples of each class are fed through the feature extractor. A mean vector based on encoded samples of each class is calculated so 4 reference vectors have been created. Next, there should be a threshold for each class to complete the similarity network. For defining these thresholds, only encoded samples from the validation and test set are compared with the reference vectors based on the euclidean distance. The maximum distance of each class is considered as its threshold. When a cropped frame is sent to the classifier, the classifier output would be compared with the references, and the smallest distance would be chosen. If the selected distance is lower than its threshold, the cropped frame belongs to the dataset so the classifier output is valid. Otherwise, it is from undesirable classes and

must be ignored, and a new frame must be captured. It can be argued that there are 4 + 1 categories 4 as mentioned in the hand dataset and 1 for other gestures, which the similarity network ignores.

Hence, in the classification part, two tasks will be done: First, the classifier predicts a label for a cropped frame. Second, the similarity network compares the cropped frame with the reference vectors and then the defined thresholds to determine whether the cropped frames represent one of the four dataset classes or not. Therefore, the classifier and similarity network act independently, and the result of the similarity network validates the predicted label. If the classification part predicts a valid label, the computer pointer will respond to that.

3.4. Pointer Commands

There should be a controller unit for allocating commands to each predicted label in order to control the computer pointer. On-command is defined in the way that if the proposed algorithm is off, it will be turned on simply by seeing the user's palm. The recognition of palm can move the pointer based on the center coordinate of its cropped frame. Since the SSD uses input images with 300 x 300 resolution, its coordinate should be converted to a meaningful coordinate for the screen. When the proposed algorithm is on, one can click or right-click where the cursor is by pointing to the left or right, respectively. By showing fists, users can turn the algorithm off. After that, no action will occur until a recognized palm turns it on again. (see Figure 3)

In figure 4, the proposed algorithm for controlling the computer cursor is visualized. Frames are captured by webcam and then imported to the hand detector. If there is a hand in a frame, the frame will be cropped, and its center coordinate would be calculated. The cropped frame is sent to classification parts. If it represents a defined gesture from the dataset, it will become a valid label and control the computer pointer.

4. Experimental Results

The proposed algorithm was developed on a personal computer using Ubuntu with GTX 1080 Ti GPU and implemented within Python programming language using Keras framework and OpenCV library. The speed of the proposed algorithm to control the pointer, such as clicking, right-clicking and moving the pointer, is 15 frames per second.

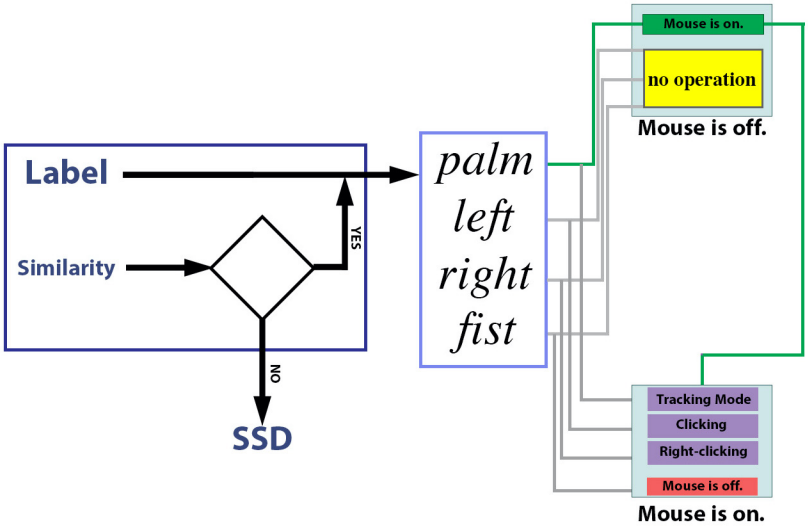

Figure 3: Designed controller unit-The pointer can move and click or right-click by receiving a valid label from the classification part.

4.1. Evaluation

For evaluating the classification part, two deep neural network architectures, VGG16 and EfficientNet-b0, are examined. Measures such as the number of parameters, test accuracy, and run-time are considered to assess the performances of each architecture. Even though VGG16 has reached higher accuracy rather than EfficientNet-B0, its parameters have been more than three times of EfficientNet-B0 parameters, so VGG16 requires more run-time as opposed to EfficientNet-B0. As a result, the classification part uses EfficientNet-B0 as its feature extractor.

As mentioned before, the training set has a different distribution from the validation and test set. Three new backgrounds, such as a white, simple, and complex background, two distances from a webcam, and two light conditions, are considered to evaluate the proposed algorithm. The three backgrounds are illustrated in Figure 6. For each situation, 10 frames including both hands are captured. As a result, 80 frames are checked for every position, and the results are presented in Table 1.

The confusion matrix of the proposed algorithm is shown in Table 2. The lowest accuracy is for clicking mode because, in the complex background, there was a brown closet that the SSD algorithm confused to detect hand gestures with that background correctly.

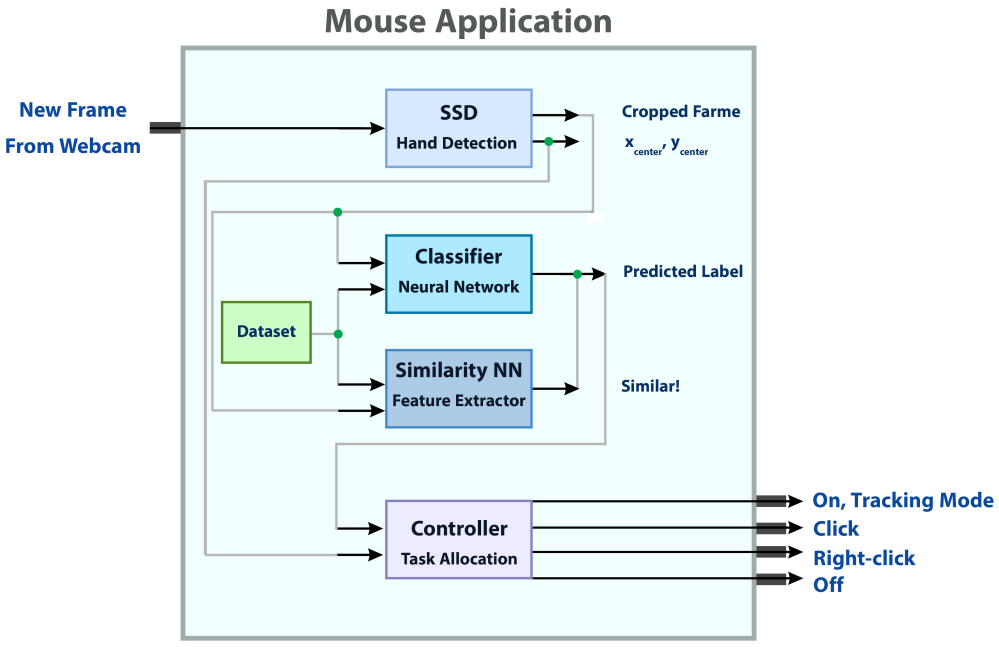

Figure 4: The proposed algorithm for controlling the computer pointer

5. Conclusion

In this paper, a human-computer interaction algorithm based on hand gesture recognition is proposed to control the computer pointer without a single contact. It is also observed that deep learning methods, such as CNN classifier and similarity network, have been used instead of the classic machine learning and image processing tools.

The proposed algorithm can be helpful for other systems in the complex background and different light conditions and camera perspectives. Since the proposed algorithm is touchless other public-place gadgets can benefit from the algorithm during the COVID-19 pandemic. Finally, the designed computer pointer controller has reached 91.88% accuracy for various backgrounds and 97% for simple backgrounds. Also, a hand gesture dataset including 6720 colored image samples with four classes is presented, which can be used for other purposes as well.

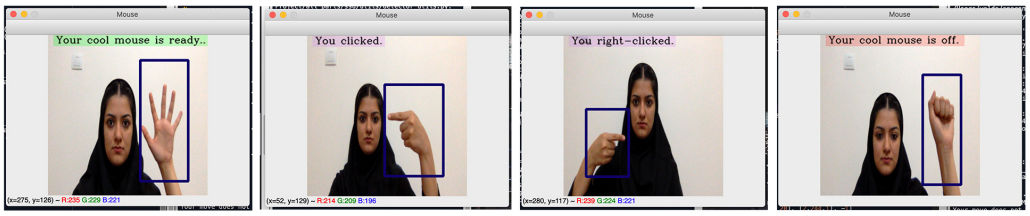

Figure 5: Controlling computer pointer by hand gesture recognition-Palm recognition results in turning on the algorithm. Pointing to the left or right contribute to click or right-click, in turn. Showing fist would turn it off as well.

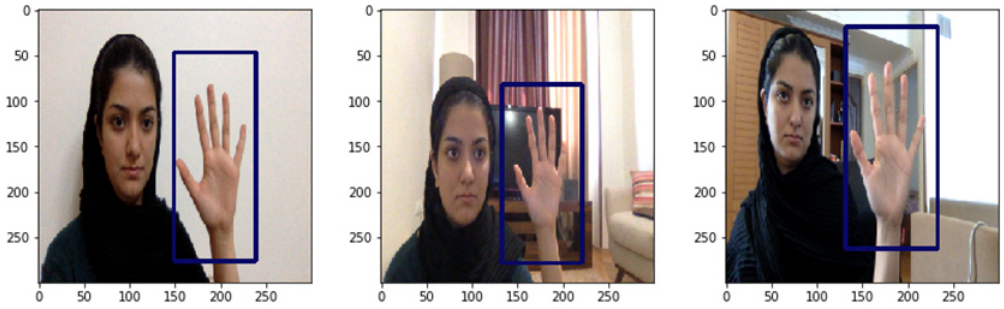

Figure 6: The chosen backgrounds for algorithm evaluation

Bibliography

[1] Manuel Gil-Martín, Rubén San-Segundo, Fernando Fernández-Martínez, and Ricardo de Córdoba. Human activity recognition adapted to the type of movement. *Computers & Electrical Engineering*, 88:106822, 2020.

[2] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE transactions on multimedia*, 15(5):1110–1120, 2013.

[3] Mohammad H Alomari, Ayman AbuBaker, Aiman Turani, Ali M Baniyounes, and Adnan Manasreh. Eeg mouse: A machine learning-based brain computer interface. *Int. J. Adv. Comput. Sci. Appl*, 5(4):193–198, 2014.

[4] Marco E Benalcázar, Cristhian Motoche, Jonathan A Zea, Andrés G Jaramillo, Carlos E Anchundia, Patricio Zambrano, Marco Segura,

Table 1: Performance of the classification part with different architectures

Network	Accuracy	Parameters	Run-time
EfficientNet-B0	99%	4.98M	671us/step
VGG16	100%	15.95M	776us/step

Table 2: Confusion matrix for each mode of controlling the cursor

	Turn Off	Turn On	Click	R-click
Turn Off	0.9125	0	0.1375	0.0375
Turn On	0.0750	0.9708	0	0
Click	0.0083	0.0250	0.8625	0.0333
R-click	0.0041	0.0041	0	0.9292

Freddy Benalcázar Palacios, and María Pérez. Real-time hand gesture recognition using the myo armband and muscle activity detection. In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, pages 1–6. IEEE, 2017.

[5] Ji-Hwan Kim, Nguyen Duc Thang, and Tae-Seong Kim. 3-d hand motion tracking and gesture recognition using a data glove. In *2009 IEEE International Symposium on Industrial Electronics*, pages 1013–1018. IEEE, 2009.

[6] Robert Y Wang and Jovan Popović. Real-time hand-tracking with a color glove. *ACM transactions on graphics (TOG)*, 28(3):1–8, 2009.

[7] Pranav Mistry and Pattie Maes. Sixthsense: a wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies: Adaptation*, pages 85–85. 2009.

[8] S Veluchamy, LR Karlmarx, and J Jeya Sudha. Vision based gesturally controllable human computer interaction system. In *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, pages 8–15. IEEE, 2015.

[9] Chetan Dhule and Trupti Nagrare. Computer vision based human-computer interaction using color detection techniques. In *2014 Fourth*

International Conference on Communication Systems and Network Technologies, pages 934–938. IEEE, 2014.

[10] Hanwen Huang, Yanwen Chong, Congchong Nie, and Shaoming Pan. Hand gesture recognition with skin detection and deep learning method. In *Journal of Physics: Conference Series*, volume 1213, page 022001. IOP Publishing, 2019.

[11] Hojoon Park. A method for controlling the mouse movement using a real time camera. *Brown University, Providence, RI, USA, Department of computer science*, 2008.

[12] Uzma Noreen, Mutiullah Jamil, and Nazir Ahmad. Hand detection using hsv model. *Hand*, 6(12), 2015.

[13] Horatiu-Stefan Grif and Trian Turc. Human hand gesture based system for mouse cursor control. *Procedia Manufacturing*, 22:1038–1042, 2018.

[14] Richa Golash and Yogendra Kumar Jain. Economical and user-friendly design of vision-based natural-user interface via dynamic hand gestures. *International Journal of Advanced Research in Engineering and Technology*, 11(6), 2020.

[15] Xiaohui Shen, Gang Hua, Lance Williams, and Ying Wu. Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields. *Image and Vision Computing*, 30(3):227–235, 2012.

[16] Globefire. globefire/hand detection tracking opencv-.

[17] Peng Liu, Xiangxiang Li, Haiting Cui, Shanshan Li, and Yafei Yuan. Hand gesture recognition based on single-shot multibox detector deep learning. *Mobile Information Systems*, 2019, 2019.

[18] Chengming Yi, Liguang Zhou, Zhixiang Wang, Zhenglong Sun, and Changgeng Tan. Long-range hand gesture recognition with joint ssd network. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1959–1963. IEEE, 2018.

[19] Zihan Ni, Jia Chen, Nong Sang, Changxin Gao, and Leyuan Liu. Light yolo for high-speed gesture recognition. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3099–3103. IEEE, 2018.

[20] Upadrasta Tanmaie and Ch Srinivasa Rao. Hand posture detection and classification using you only look once (yolo v2) object detector.

[21] Kianoush Haratiannejadi, Neshat Elhami Fard, and Rastko R Selmic. Smart glove and hand gesture-based control interface for multi-rotor aerial vehicles. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1956–1962. IEEE, 2019.

[22] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[23] Erfan Ashtari, Mohammad Amin Basiri, Saeid Mohammadi Nejati, Hemen Zandi, Seyyed Hossein SeyyedAghaei Rezaei, Mehdi Tale Masouleh, and Ahmad Kalhor. Indoor and outdoor face recognition for social robot, sanbot robot as case study. In *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, pages 1–7. IEEE, 2020.