Development of a Hand Gesture Based Control Interface Using Deep Learning

Dennis Núñez-Fernández(✉)

Universidad Nacional de Ingeniería, Lima, Peru
dnunezf@uni.pe

Abstract. This paper describes the implementation of a control system based on ten different hand gestures, providing a useful approach for the implementation of better user-friendly human-machine interfaces. Hand detection is achieved using fast detection and tracking algorithms, and classification by a light convolutional neural network. The experimental results show a real-time response with an accuracy of 95.09%, and making use of low power consumption. These results demonstrate that the proposed system could be applied in a large range of applications such as virtual reality, robotics, autonomous driving systems, human-machine interfaces, augmented reality among others.

Keywords: Gesture recognition · Human-machine interface · Deep learning · Real-time · Hand poses

1 Introduction

Hand gesture recognition is an important way to build user-friendly human-machine interfaces. In a near future, hand posture recognition technology would allow for the operation of complex machines and smart devices through only series of hand postures, finger and hand movements, eliminating the need for physical contact between user and machine. However, hand poses recognition on images from single camera is a difficult task due occlusion, variations of posture, appearance and differences in hand shapes. Despite these problems, several approaches to gesture recognition on images has been proposed [1].

During the last years, Convolutional Neural Networks (CNNs) have become the state-of-the-art for object recognition tasks [2–4]. However, only few papers report successful results [1,5]. Some obstacles to wider and more efficient use of CNNs in hand pose classification problem are lack of sufficiently large datasets, high computational costs, as well as lack of hand detectors suitable for CNN-based classifiers. In [6], a CNN has been used for classification of 6 hand poses to control robots via colored gloves, but the use of such additional hardware makes of the system difficult to employ for touchless applications. A more recent work [7], a multichannel CNN for the Nao humanoid robot was implemented, it employs the JTD dataset and make use of three channels, they obtained an F1 score of 92%. In another recent work [8], a CNN was trained on one million of

J. A. Lossio-Ventura et al. (Eds.): SIMBig 2019, CCIS 1070, pp. 143–150, 2020.
https://doi.org/10.1007/978-3-030-46140-9_14

images, however only a portion of the dataset with 3361 manually labeled frames in 45 classes of sign language is publicly available, it makes of such work difficult to reproduce. In addition, state-of-the-art works [9] obtain a high accuracy rate but use depth cameras and large CNNs, which make difficult to use in human-machine interfaces that demand a real-time response.

In this work we developed a system for hand pose recognition to work on embedded computers with limited computational resources. In order to accomplish the targets, we employ low-processing algorithms and a light CNN, which was optimized to balance high accuracy, high response time and low power and computational consumption.

2 Proposed Method

2.1 Overview

The proposed system works with images captured from a standard camera and executed on a regular computer with low computational resources without GPU support. Therefore, the main objectives are as follows: high accuracy rate, fast time response, low power consumption and low computational costs.

The system is composed of three main steps: hand detection, hand region tracking and hand gesture recognition (see Fig. 1). In the first step the Haar cascades classifier detects a basic hand shape in order to have a good hand detection. Then, this hand region is tracked using the MOSSE (Minimum Output Sum of Squared Error) tracking algorithm. Finally, hand gesture recognition is performed based on a trained Convolutional Neural Network. Since the steps described before are designed to consume few computational resources, the whole system will be implemented on a personal computer without GPU support.

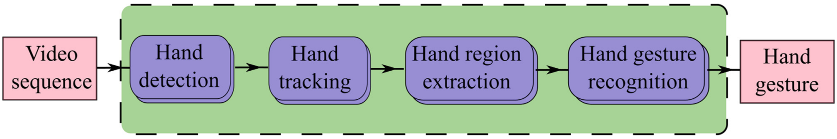


Fig. 1. Diagram for the proposed recognition system

2.2 Hand Detection and Tracking

Haar cascade classifier allows better detection of objects with static features such as balloons, boxes, faces, eyes, mounts, noise, etc. But a hand in motion has few static features because its shape and fingers can change as well as its orientations. So, Haar cascade classifier allows detection of only basic hand poses, which are not suitable to recognize a hand in motion with a long mount of different poses.

Since hand detection using Haar cascades in not a robust method, this deficiency is compensated with a hand tracker based on wrist region. This hand region is proposed for tracking due this region have both invariant and static features when hand changes to different poses, shapes and orientations.

In addition to this, hand tracking allows the reduction of the processing time since tracking requires less computational resources than hand detection (whole image evaluation versus local evaluation). Figure 2 shows the different hand regions used for detection and tracking, as image shows the hand region for tracking (blue box) encloses the hand in different shapes and poses. Therefore, the hand region inside the blue box will be used by the CNN to perform gesture recognition.

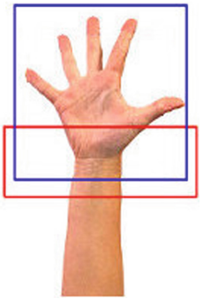


Fig. 2. Wrist region for detection (red) and hand region for tracking (blue) (Color figure online)

In this project, the MOSSE tracking (Minimum Output Sum of Squared Error) algorithm is used for hand tracking. The MOSSE tracker uses adaptive correlation for object tracking which produces stable correlation filters when initialized using a single frame [10]. MOSSE tracker is robust to variations in lighting, scale, pose, and non-rigid deformations. It also detects occlusion based upon the peak-to-sidelobe ratio, which enables the tracker to pause and resume where it left off when the object reappears. MOSSE tracker also operates at a higher fps (450 fps and even more). In addition, the proposed tracking algorithm consumes less memory and computational resources than the Haar cascade classifier.

2.3 Skin Detection

Skin color is a potent characteristic for fast hand region detection. Essentially, all skin color-based methodologies try to learn a skin color distribution, and then use it to extract the hand region. In this project the hand region has been obtained on the basis of statistical color models [11]. In this way, a model in RGB-H-CbCr color spaces were constructed on the basis of a training set. Later, the hand probability image was threshold. Finally, after morphological closing, a connected components labeling was used to extract the gravity center of the hand region, the coordinates of the most top pixel as well as coordinates of the most left pixel of the hand region.

2.4 Hand Poses Dataset

The dataset for hand gesture classification was obtained from a publicly available database of the AGH University of Science and Technology. It is composed of 73,124 grayscale images of size 48 × 48 pixels divided into 10 different hand poses, captured from different persons of various nationalities. This dataset was divided in 80% (42,027 images) for training and 20% (14,667 images) for testing. Figure 3 shows samples of each hand gesture, also called class. The principal benefit of this dataset is that in each class the wrists are approximately located at the same position. Furthermore, thanks to such an approach the recognition of hand poses at acceptable frame rates can be succeeded with a simple convolutional neural network and at a lower computational cost.

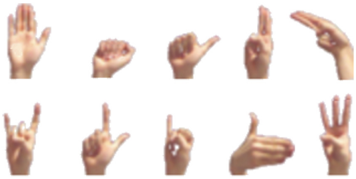


Fig. 3. Hand gestures poses

2.5 Convolutional Neural Network

For the CNN we use binary images of 48 × 48 pixels, and a small CNN with fewer layers and learnable parameters. The proposed CNN consists of two convolutional layers with kernels of 5 × 5 size each one, a non linearity (ReLU) activation function and a max-pooling layer after every convolutional layer, and two full-connected (FC) layers of 150 neurons length followed by a final 10-way softmax (see Fig. 4). Additionally, the posposed CNN has only 60 K learnable parameters. This number of parameters are significantly less than the AlexNet network (60 M learnable parameters) [2] and the GoogleNet (6.8 M learnable parameters) [12].

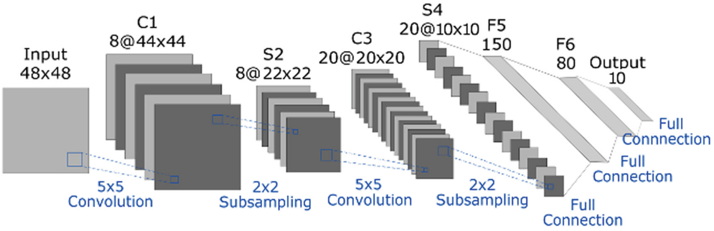
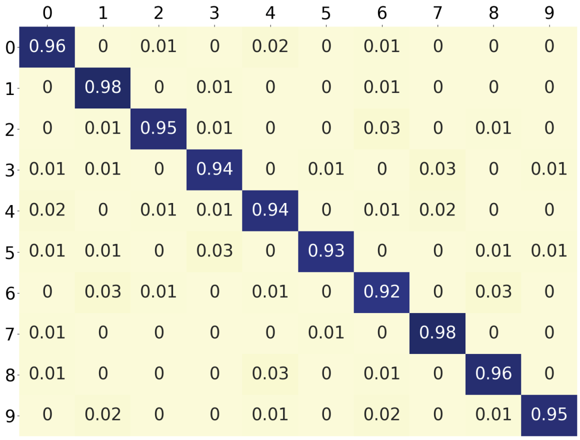


Fig. 4. Architecture of the proposed CNN

3 Experimental Results

3.1 Experimental Results of the Model

The performance of the CNN for hand poses classification was evaluated using different metrics such as the accuracy and the confusion matrix. This matrix presents a visualization of the misclassified classes and helps to add more training images in order to improve the model. The confusion matrix of our model is shown in Fig. 5 and discloses which hand poses are misclassified. These errors happen because of similarities between the classes. Furthermore, our architecture shows an outstanding accuracy of 95.09% and a F1 score of 95.12%.

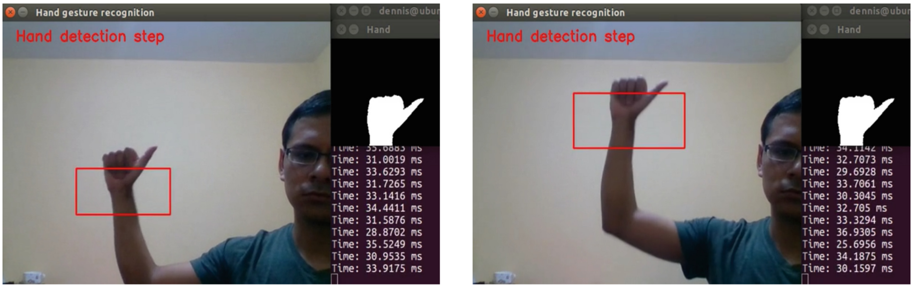


	0	1	2	3	4	5	6	7	8	9
0	0.96	0	0.01	0	0.02	0	0.01	0	0	0
1	0	0.98	0	0.01	0	0	0.01	0	0	0
2	0	0.01	0.95	0.01	0	0	0.03	0	0.01	0
3	0.01	0.01	0	0.94	0	0.01	0	0.03	0	0.01
4	0.02	0	0.01	0.01	0.94	0	0.01	0.02	0	0
5	0.01	0.01	0	0.03	0	0.93	0	0	0.01	0.01
6	0	0.03	0.01	0	0.01	0	0.92	0	0.03	0
7	0.01	0	0	0	0	0.01	0	0.98	0	0
8	0.01	0	0	0	0.03	0	0.01	0	0.96	0
9	0	0.02	0	0	0.01	0	0.02	0	0.01	0.95

Fig. 5. Confusion matrix

3.2 Experimental Results of Inference

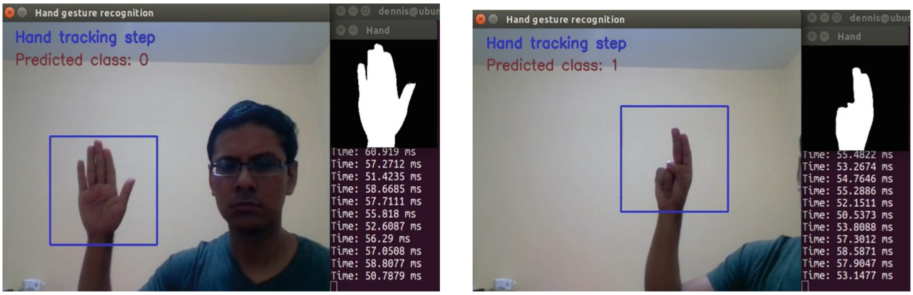
The implementation of the proposed recognition system on a personal computer with GPU support has no issues due to its high computational resources. However, when a recognition system is implemented on standard computers with no GPU support we have two major obstacles working against us: limited RAM memory and restricted processor speed. In order to obtain a better computation performance, the system was implemented using C++ language. In spite of the processing and memory limitations mentioned above, our real-time recognition system shows promising results during the evaluation step. Figure 6 depicts its performance under real conditions. As you can see, the system correctly recognizes different hand poses, despite different shifted positions, shape distortions, low light conditions, different sizes, and even when the recognition is done on images taken by a different camera. In addition, we obtain a fast response time of about 55.1 ms (average of 100 iterations) to detect and classify a single hand pose (Fig. 7).



```
Time: 35.6883 ms
Time: 31.0019 ms
Time: 33.6293 ms
Time: 31.7265 ms
Time: 33.1416 ms
Time: 34.4411 ms
Time: 31.5876 ms
Time: 28.8702 ms
Time: 35.5249 ms
Time: 30.9535 ms
Time: 33.9175 ms
```

```
Time: 34.1142 ms
Time: 32.7073 ms
Time: 29.6928 ms
Time: 33.7061 ms
Time: 30.3045 ms
Time: 32.705 ms
Time: 33.3294 ms
Time: 36.9305 ms
Time: 25.6956 ms
Time: 34.1875 ms
Time: 30.1597 ms
```

Fig. 6. Results of hand pose detection on a personal computer



```
Time: 60.919 ms
Time: 57.2712 ms
Time: 51.4235 ms
Time: 58.6685 ms
Time: 57.7111 ms
Time: 55.818 ms
Time: 52.6087 ms
Time: 56.29 ms
Time: 57.0508 ms
Time: 58.8077 ms
Time: 50.7879 ms
```

```
Time: 55.4822 ms
Time: 53.2674 ms
Time: 54.7646 ms
Time: 55.2886 ms
Time: 52.1511 ms
Time: 50.5373 ms
Time: 53.8088 ms
Time: 57.3012 ms
Time: 58.5871 ms
Time: 57.9047 ms
Time: 53.1477 ms
```

Fig. 7. Results of hand pose tracking and classification on a personal computer

Table 1 shows some details of CNNs tested on the personal computer without GPU support. As the table shows, the proposed CNN achieves a fast response time of around 60 ms and a low power consumption of 0.47 Joules per image. These results were expected due to their simple but efficient design explained in the previous sections. Furthermore, compared with state-of-the-art CNN models for similar recognition tasks and tested on an identical computer without GPU support, the proposed architecture achieves the highest response time and the lowest power consumption.

Table 1. Response time and power consumption for evaluation of different CNN models on a personal computer without GPU support using Caffe framework

Model	Proposed CNN	AlexNet v2 [13]	OverFeat [13]	VGG_A [13]	GoogleNet [13]
Layers	8	11	11	16	22
Power (J./img.)	0.47	0.75	2.00	5.00	3.50
Time (s.)	0.06	0.35	0.88	2.35	1.87

4 Conclusions

This paper introduced the implementation of a hand gesture recognition system for a touchless control interface based on images acquired by a single color camera. In order to obtain the fastest response time and a low power consumption, we employed fast computer vision algorithms and a light convolutional neural network. In this way, the proposed recognition system shows promising results, achieving an accuracy of 95.09% for the classification of 10 different hand poses. Furthermore, the evaluation of the hand gesture system in a personal computer with a single image gives an average processing time of about 55.1 ms, and a low energy consumption of 0.47 J. per image. The results mentioned above demonstrate that the proposed recognition system can be used in a large range of applications, from robotics to entertainment.

References

1. Oyedotun, O.K., Khashman, A.: Deep learning in vision-based static hand gesture recognition. Neural Comput. Appl. **28**(12), 3941–3951 (2016). https://doi.org/10.1007/s00521-016-2294-8
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
3. Kwolek, B.: Face detection using convolutional neural networks and gabor filters. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 551–556. Springer, Heidelberg (2005). https://doi.org/10.1007/11550822_86
4. Arel, I., Rose, D., Karnowski, T.: Research frontier: deep machine learning-a new frontier in artificial intelligence research. Comp. Intell. Mag. **5**(4), 13–18 (2010)
5. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans. Graph. **33**(5), 1–10 (2014)
6. Nagi, J., Ducatelle, F., et al.: Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: IEEE ICSIP, pp. 342–347 (2011)
7. Barros, P., Magg, S., Weber, C., Wermter, S.: A multichannel convolutional neural network for hand posture recognition. In: Wermter, S., Weber, C., Duch, W., Honkela, T., Koprinkova-Hristova, P., Magg, S., Palm, G., Villa, A.E.P. (eds.) ICANN 2014. LNCS, vol. 8681, pp. 403–410. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11179-7_51
8. Koller, O., Ney, H., Bowden, R.: Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3793–3802 (2016)
9. Yuan, S., Ye, Q., Stenger, B., Jain, S., Kim, T.: BigHand2.2M benchmark: hand pose dataset and state of the art analysis. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 2605–2613 (2017)
10. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, pp. 2544–2550 (2010)

11. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. Int. J. Comput. Vision **46**(1), 81–96 (2002)
12. Szegedy, C., et al.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA (2015)
13. Li, D., Chen, X., Becchi, M., Zong, Z.: Evaluating the energy efficiency of deep convolutional neural networks on CPUs and GPUs. In: 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom), Atlanta, GA, pp. 477–484 (2016)