# Tiny Hand Gesture Recognition without Localization via a Deep Convolutional Network

Peijun Bao, Ana I. Maqueda, Carlos R. del-Blanco, and Narciso García

***Abstract*— Visual hand-gesture recognition is being increasingly desired for human-computer interaction interfaces. In many applications, hands only occupy about 10% of the image, whereas the most of it contains background, human face, and human body. Spatial localization of the hands in such scenarios could be a challenging task and ground truth bounding boxes need to be provided for training, which is usually not accessible. However, the location of the hand is not a requirement when the criteria is just the recognition of a gesture to command a consumer electronics device, such as mobiles phones and TVs. In this paper, a deep convolutional neural network is proposed to directly classify hand gestures in images without any segmentation or detection stage that could discard the irrelevant not-hand areas. The designed hand-gesture recognition network can classify seven sorts of hand gestures in a user-independent manner and on real time, achieving an accuracy of 97.1% in the dataset with simple backgrounds and 85.3% in the dataset with complex backgrounds.**

***Index Terms*—Deep learning, hand gesture recognition, human-machine interface, mobile phone, neural network, no localization, TV.**

## I. Introduction

HAND gesture recognition plays an important role in human-computer interaction (HCI) for touchless interfaces [1][2][4]. In many vision based applications, the semantic of the hand gesture is independent on its specific location [1][3]. However, it is a common practice to localize the hand inside the image to discard the irrelevant areas that could complicate the inference process of the hand gesture, especially when the hand only occupies a small part of the image (see Fig. 1). Many hand gesture recognition systems follow this approach. For example, a universal remote control system using hand gestures is presented in [5], where the hands are segmented from the background by using motion and color skin information. Similarly, the color skin information is used to segment the hands in [6] for another generic hand-based remote control system, although in this case the camera is activated by a Pyroelectric Infrared (PIR) sensor to save energy consumption. In [7], a visual hand gesture interface for TVs is proposed, where the hand location is inferred from the trajectory information computed via a particle filter framework.

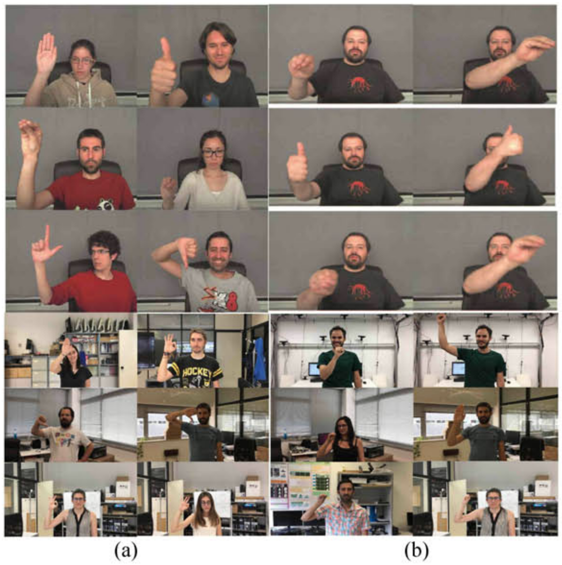


Fig. 1. (a): hand gestures only occupy a small part in the image. (b): examples of changes in hand position, shape, and scale.

Other examples are based on deep learning techniques that have proven a superior performance during the last years. In [12] the hand localization is jointly performed with the hand gesture recognition task, using a multi-resolution sliding window that densely selects image regions, and evaluates for each selection if a specific hand gesture is performed. Alternatively, some works [8][9][10][11] first propose a reduced set of image regions that potentially can contain hands, and then send those regions to a classification system for the proper hand gesture inference. Region proposal based algorithms are more computationally efficient in dealing with hands of different sizes and locations than those based on a multi-resolution sliding window. On the other hand, the proposed regions usually have different sizes, and therefore they need to be warped to a canonical size for the classification system, which results in unwanted distortions that can compromise the accuracy [13].

Commonly for both approaches, ground truth in the form of bounding boxes containing hands (besides the own hand gesture label) should be provided for training the classification system. This is particularly important since it is far more difficult to collect hand bounding boxes than hand gesture labels, due to the involved heavy manual labor to annotate the bounding boxes.

In this paper, a novel hand-gesture recognition system is proposed. It is able to directly recognize a hand gesture from the whole image without using any region proposal algorithm or sliding window mechanism, and therefore facing the problem of recognizing a pattern with relatively small size and random location embedded in clutter (upper body and background). The recognition system is based on a deep convolutional neural network (CNN) that can effectively recognize hand gestures in arbitrary positions that occupy a reduced image area in comparison with the surrounded background (including the own human face and body). For this purpose, one of the main guidelines for the design of CNN has been to deal with the potential overfitting problems that arise in such conditions [15][16], which usually derive in that the deep CNN only memorize the training data, and is unable to generalize to new data. Other major advantage of our system is that no bounding boxes are required for the training stage, which simplifies the collection of the training samples and the own training procedure. Moreover, the hand-gesture recognition system can operate in real-time because of the lack of the highly computational stages of region proposal or sliding window. Fig. 2 illustrates the two different methods to perform hand gesture recognition: with and without hand region proposal.

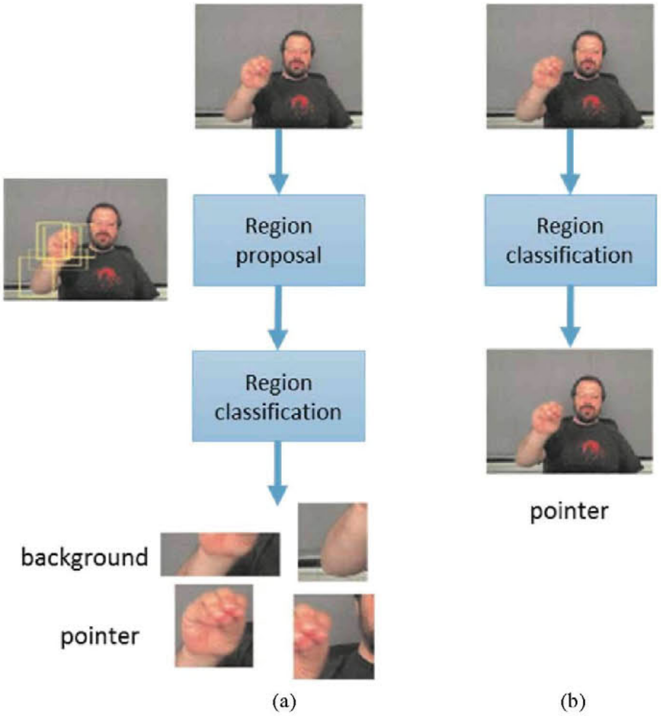


Fig. 2. Two different methods to perform hand gesture recognition. (a) Hand gesture recognition with region proposal. (b) Hand gesture recognition without region proposal.

## II. Related work

### A. *Hand Gesture Recognition*

Different features have been proposed for the recognition of hand gestures, such as motion and skin color [5][6], hand-crafted spatio-temporal descriptors [20], articulated models, and trajectory based information [7]. These features are then processed by an inference algorithm that recognizes the specific hand gesture, such as Hidden Markov Models (HMM) [17], Conditional Random Fields (CRF) [19], Support Vector Machines (SVM) [21][22], and Convolutional Neural Networks (CNN) [12]. All of them have in common the localization of the hand in order to recognize the gesture. For example, the work presented in [22] describes a hand gesture recognition system that includes the following stages: hand segmentation for localization, feature extraction using a 2D shape descriptor, and classification based on SVMs.

### B. *Deep Convolutional Neural Networks*

Recently, there has been a rapid and substantial development on deep convolutional neural networks. Most of state-of-the-art works that currently deal with classification, detection, and segmentation tasks are adopting the design of a deep CNN. For instance, deep CNN have been successfully applied to applications such as pedestrian detection [24], face recognition [25], and hand-gesture recognition [12][23]. Focusing on hand gestures, the work presented in [12] introduces a 3D convolutional neural network to recognize dynamic hand gestures in video sequences where the hand occupies most of the image. The classification system consists of two sub-networks: a high-resolution network and a low-resolution one, each one containing four 3D-convolutional layers, four max-pooling layers, and three fully-connected layers. The two networks are then combined by multiplying element-wise their respective class-membership probabilities.

### C. *Small Object Recognition via Deep CNN*

The Region-based convolutional neural network (R-CNN) approach [8][9][10] is an effective system to detect and classify small objects among complicate backgrounds in an image. Selective search is performed via a region proposal algorithm, extracting about 2,000 region proposals before the system warps these regions of different size to one canonical size. Then, it computes CNN features for each region proposal and classifies them via a SVM. However, the system cannot operate in real-time because of the heavy processing related with CNN feature computation and classification of the region proposals. At test-time, the hand gesture recognition takes more than 40 seconds per image. Moreover, the accuracy of classification is limited due to its dependence on the performance of the region proposal module and the distortion caused by the warping of each image region.

Based on R-CNN, Faster R-CNN introduces a Region Proposal Network (RPN) as region proposal algorithm [10]. This algorithm uses a CNN to predict both foreground pixels on a coarse grid and the coordinates of the corresponding bounding box. The RPN can predict high-quality region-proposals, sharing neural network parameters with the other CNN used for the classification. Faster R-CNN can achieve compelling accuracy and speed. One requirement is that bounding boxes should be provided for training the RPN.

An alternative approach is to adopt a sliding window based detector [23][24][25]. CNN has been used in this way for at least two decades, especially for constrained object categories, such as faces, hands, and pedestrians. However, to reach a reasonable computational efficiency is necessary that the different appearances of the objects to be recognized share a common aspect ratio, which is not the case for hand gesture recognition. The aspect ratio problem can be addressed with mixture models, where each component specializes in a narrow band of aspect ratios, or even with bounding box regression. However, the associated computational cost increases.

## III. Hand gesture recognition system

The proposed system is able to directly recognize hand gestures from the whole image without using any image region selection framework. This is accomplished by a careful design of a deep CNN that can successfully recognize hand gestures performed in any location inside the image under the challenging situation that the hand that performs the gesture only takes up a reduced image area in comparison with the whole image. Even more, the image is cluttered by background objects, among others human faces and upper bodies. As a consequence, there are two main guidelines that have oriented the design of the developed deep CNN: hand region shrinking and overfitting.

The next sub-sections describe the developed deep CNN design, starting by how a CNN can detect patterns embedded in arbitrary locations in an image, continuing by addressing the region shrinking and the overfitting design restrictions, and ending with the description of the proposed CNN.

### A. Detection of embedded patterns in an image

A CNN has a convolutional structure that restricts the neural connection between layers (acting as local receptive fields) and forces to share the same weights inside a layer. Therefore, all the neurons involved in convolutional filter operations at one layer detect exactly the same feature, but at different localizations in the input image. I.e., the filters are invariant to the translation of patterns inside the image. Namely, Conv(X + X0, W) = Conv(X, W) + X0, where X is the image, W is the filter, X0 is a translation, and Conv is the convolutional operation. Fig. 3 illustrates this mechanism, showing the output feature maps of a convolutional layer and some final detections.

### B. Hand region shrinking

To get a high performance recognition system in a scenario where the pattern to be detected (the hand) is relatively small in comparison with the image size, it is very important to extract high-level abstract features from raw data. It is well-known that deeper networks allow to learn more powerful semantic representations. Therefore, the number of convolutional layers and pooling layers should be large enough for the ongoing recognition task. However, the reduced size of the hand inside the image produces a shrinking problem according the data travel deeper and deeper across the network. The effective activated area containing relevant hand information for classification task shrinks exponentially with the increasing number of layers. For example, setting a stride of 3 pixels for a two layer network composed by a convolutional and a pooling layer, the area of the output feature map would be reduced by a factor of 9. This behavior would produce that the information coming from the small hand regions would be dissolved by the surrounding background information in a few layers. To keep a high classification performance, a reasonable ratio between the feature map area and the hand gesture area should be kept. To enforce such condition, the following criteria is adopted in the CNN design: the stride values for the convolutional layers are set to only 1 (except for the first layer to improve the computational efficiency), and the stride values for the max-pooling layers are set to 2. When a stride value of 1 is set for the convolutional layers, there is no evident region shrinking. Thus, the area of the image is only reduced when max-pooling operation is used. Therefore, there should be multiple convolutional layers before each max-pooling layer in order to extract enough features for the classification, preventing an excessive hand region shrinking.

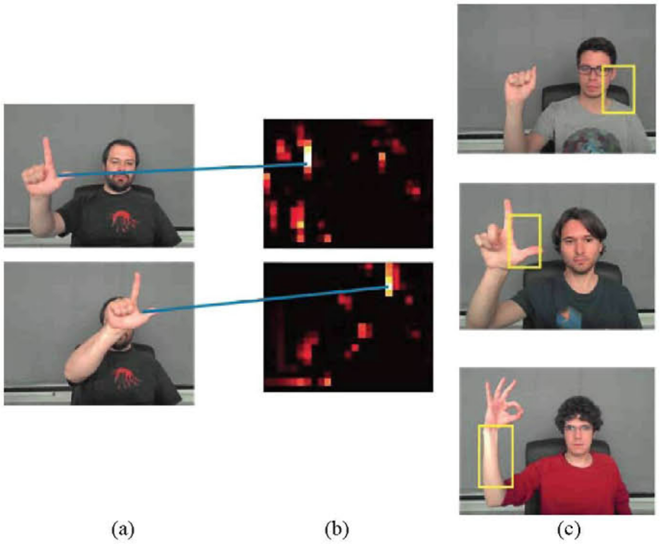


Fig. 3. Detection of the same feature at different locations. Fig. 3(a) shows two images containing L-like shape hand gestures at different locations. Fig. 3(b) shows the corresponding feature maps generated by a convolutional filter associated to one layer of the neural network. Fig. 3(c) shows more examples where the strongest activation of the filter is detecting the L-like shape.

### C. Overfitting

Overfitting is a common problem in machine learning [15][16]. For the proposed hand recognition task, the region of interest is relatively small, causing misleading behaviors in the CNN learning, such as trying to infer the hand gesture from non-related image areas (for example the human body or the human face). Therefore, overfitting becomes a major problem in such conditions. For preventing this kind of behaviors, the

CNN design should restrict the number of network parameters. For this purpose, small convolutional filter sizes of 3x3 pixels have been adopted, except for the first layer that is 5x5 pixels. This leads to a significant improvement in the accuracy in comparison with larger filter sizes. Additionally, a dropout learning strategy [28] has been applied to the fully connected layers [26] with the same purpose of preventing overfitting. The dropout strategy randomly sets zero values to the network neuron's outputs to prevent the undesired effect of vanishing gradients. The network includes two dropout layers with a dropout ratio of 0.5 (50% probability of dropping the units along their connections) during training, which prevents units from co-adapting too much. This technique could be thought as an ensemble method via the sampling from a large number of different thinned networks. Finally, an early stopping criteria for the training has been adopted as soon as the performance on the validation data starts to decrease, which complements the set of measures oriented to avoid the overfitting.

### *D. Architecture*

The proposed deep CNN design is composed by 9 convolutional layers, 4 pooling layers, 3 fully connected layers, interlaced with ReLU (Rectified Linear Unit) and dropout layers (see Fig. 4).

The first convolutional layer has a size of 5x5 and uses a stride of 3 for the sake of the computational efficiency. Consider as example an input image of 320x240 components, then the resulting output of the first convolutional layer would have a size of 106x79 components, which is much smaller than the input. This reduces the amount of processing in the following layers, making the CNN more affordable in both computational efficiency and memory requirement. The rest of convolutional layers have a size of 3x3 and use a stride value of 1, while the pooling layers have a stride value of 2 (as indicated in section III.B).

Next, a first fully connected layer is used that receives an output of 64x7x5 elements and contains 128 neurons. This is followed by a ReLU and a dropout layer. Then, a second fully connected layer is connected, receiving the previous 128-dimensional output of the first fully connected layer, and again containing 128 neurons. This is similarly followed by a ReLU and a dropout layer. Next, a third fully connected layer is used, which computes the mapping to the final classes for hand gesture recognition.

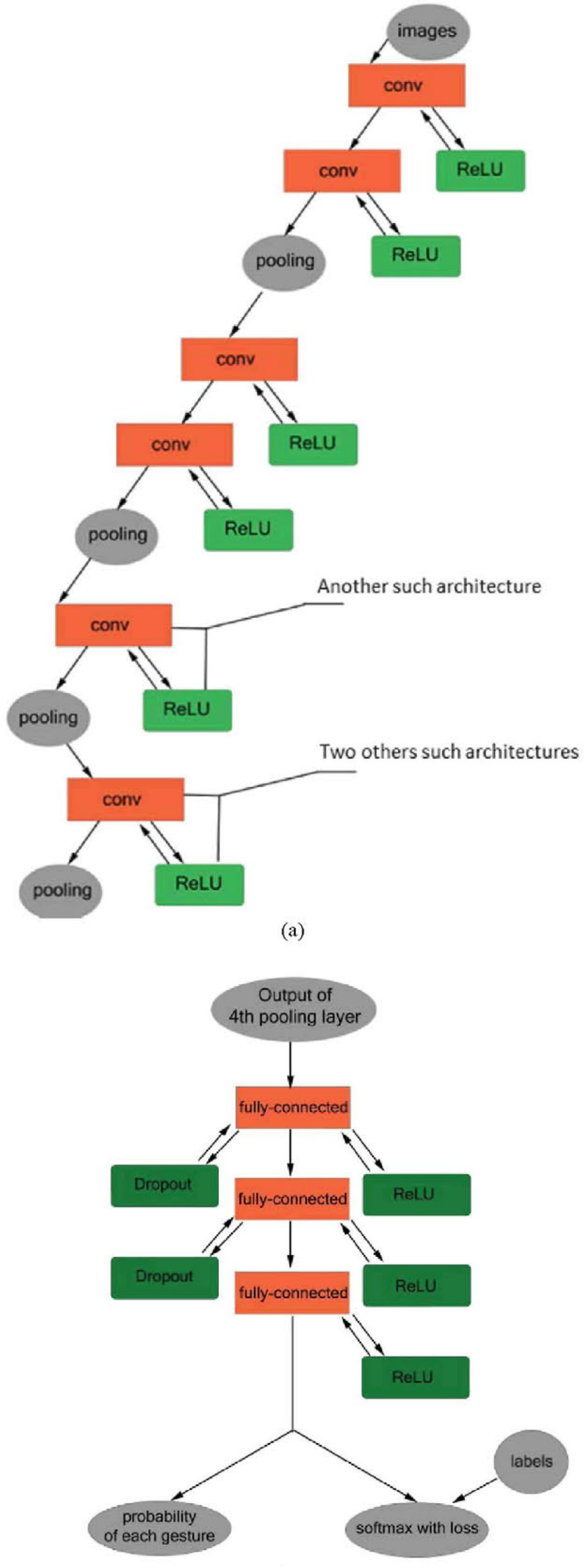
Finally, the output of the last fully connected layer is fed to a soft-max layer that assigns a probability for each class. The prediction itself is made by taking the class with maximal probability for the given input image.

## IV. Experiments

The proposed hand gesture recognition system based on a deep CNN has been implemented using an open-source framework [27]. Training was performed on a GPU with 1664 cores, base clock of 1050MHz, and a memory of 4 GB.

### *A. Dataset Preparation*

A dataset [28] has been collected from 40 people, each with 7 different hand gestures. Half of people have been recorded with complex backgrounds and the rest of them with simple backgrounds. The considered complex backgrounds are highly cluttered and the illumination undergoes large variations (see



(a)

(b)

Fig. 4. Architecture of the proposed deep CNN for hand gesture recognition separated in two parts, where (a) is the first one, and (b) the second part.

Fig. 1). Every instantiation of one gesture is composed by about 1,400 frames, and the gestures are performed in different locations in the image. To augment data, synthetic translations have been performed over the whole images, obtaining a total number of 500,000 hand gesture samples for training. For each image, human face and human body are the main parts of the image, while the hand gesture to be classified only occupies the 10% of pixels the whole image. To get a reliable result, people who appear in the testing set are totally different from the training set. Cross-validation strategies are adopted, each one with 25 people for training, 5 person for validation, and 10 people for testing.

### *B. Image Preprocessing*

The convolutional layers before the first pooling layers are the bottleneck of the computational efficiency and memory requirements. In order to make a real-time classification, the 640 x 480 images are downsampled to 320 x 240, and converted to gray scale. Thus, the number of parameters in the first convolutional layer drops 12 times than the one without the preprocessing. The proposed network has been also applied to the original RGB images, resulting in no evident improvement in performance, but with a slower speed in both training phase and testing phase.

### *C. Results*

Network training is quite fast, requiring only ~50 minutes on GPU. Hand gesture classification on a single image using the proposed network requires about 2.96 milliseconds (ms) on GPU. Classification running times can be substantially improved by running the network on images batches (requiring 0.73 ms per image with a batch size of 256).

Table I and II presents the confusion matrix for the hand gesture classification using the dataset in [28]. In general, the deep CNN can classify the hand gesture accurately, with average accuracy of 97.1% in the images with simple backgrounds and 85.3% in the images with complex backgrounds.

TABLE I
CONFUSION MATRIX FOR HAND GESTURE RECOGNITION ON SIMPLE BACKGROUNDS

| class | fist | l | ok | palm | pointer | down | up |
|---|---|---|---|---|---|---|---|
| Fist | **0.993** | 0.01 | 0 | 0 | 0 | 0 | 0 |
| l | 0.001 | **0.994** | 0.004 | 0 | 0 | 0 | 0.002 |
| ok | 0 | 0 | **0.969** | 0.015 | 0 | 0 | 0.016 |
| palm | 0 | 0 | 0 | **0.995** | 0.005 | 0 | 0 |
| pointer | 0 | 0 | 0.001 | 0 | **0.999** | 0 | 0 |
| down | 0 | 0 | 0 | 0 | 0.002 | **0.998** | 0 |
| up | 0 | 0 | 0 | 0 | 0.008 | 0.001 | **0.991** |

Table III shows the comparison result of the proposed model with AlexNet [29] and VGG19 [30] in the images with simple backgrounds and complex backgrounds. The proposed model outperforms AlexNet in both the computational time and the average accuracy. On the other hand, VGG19 has a similar accuracy with the proposed network in images with simple backgrounds, but the accuracy is worst in complex backgrounds. And also, it spends more time to process an image.

TABLE II
CONFUSION MATRIX FOR HAND GESTURE RECOGNITION ON COMPLEX BACKGROUNDS

| class | fist | l | ok | palm | pointer | down | up |
|---|---|---|---|---|---|---|---|
| Fist | **0.848** | 0.079 | 0.004 | 0.010 | 0.049 | 0.004 | 0.004 |
| l | 0.062 | **0.879** | 0.014 | 0.011 | 0.011 | 0.001 | 0.024 |
| ok | 0.025 | 0.075 | **0.759** | 0.066 | 0.017 | 0.001 | 0.056 |
| palm | 0.003 | 0.002 | 0.003 | **0.976** | 0.012 | 0.001 | 0.004 |
| pointer | 0.152 | 0.029 | 0.001 | 0.032 | **0.762** | 0.002 | 0.019 |
| down | 0.013 | 0.001 | 0.000 | 0.005 | 0.001 | **0.866** | 0.115 |
| up | 0.014 | 0.073 | 0.003 | 0.034 | 0.021 | 0.048 | **0.806** |

TABLE III
COMPARISON RESULT

| Measure | Proposed network | AlexNet | VGG19 |
|---|---|---|---|
| Accuracy on Simple Backgrounds | 97.1 | 86.3 | 96.2 |
| Accuracy on Complex Backgrounds | 85.3 | 69.4 | 77.6 |
| GPU time (ms/per image) | 2.96 | 4.89 | 25.10 |

Fig. 5 shows some notable example of correct classification. The size and shape of the gestures in Fig. 5(a) and (b) differ significantly, and despite this fact the system can effectively predict correct classifications. In Fig. 5(c), the illumination and resolution have evident changes, but the system proves to be still robust.

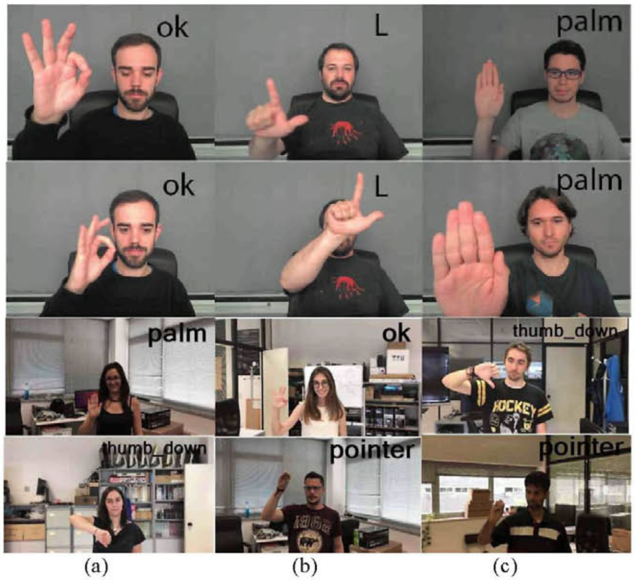


Fig. 5. Notable examples of correct classification

Fig. 6 shows some misclassifications. They are mistakes caused by the simultaneous appearance of a second hand in the image. Mistakes can also occurs when a part of a hand gesture is cropped, and therefore invisible.

## V. CONCLUSIONS

Differently from many previous methods that address the hand gesture recognition using hand localization information, the proposed deep CNN can classify hand gestures from the

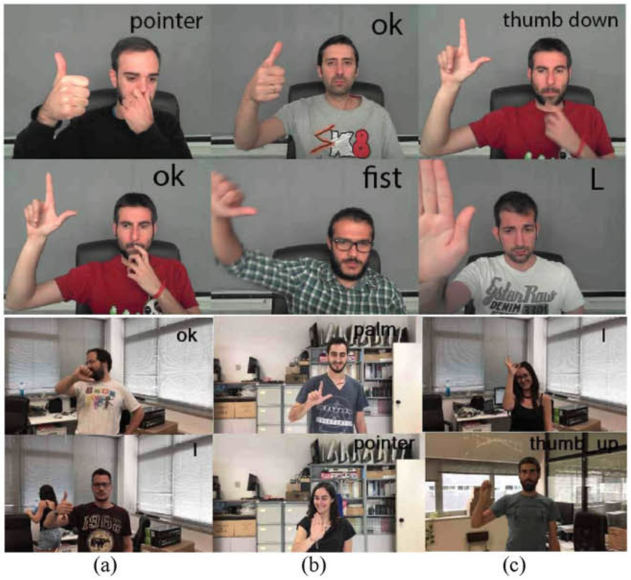


Fig. 6. Examples of misclassification

whole image without any hand localization information. The deep CNN does not rely on any region proposal algorithm or sliding window strategy, favoring the real-time operation. This strategy has also significant advantages for the creation of the training database and the own learning procedure, since no bounding boxes specifying hand regions are necessary. The presented approach is very appealing for the practical and real-time recognition of gestures that can command a consumer electronics device, such as mobiles phones and TVs, since it is more affordable and efficient than other approaches both in speed and memory.

## REFERENCES

[1] S.H. Lee, M.K. Sohn, D.J. Kim, B. Kim, and H. Kim, "Smart TV interaction system using face and hand gesture recognition," in *Proc. ICCE*, Las Vegas, NV, 2013, pp. 173-174.

[2] S. Kim, G. Park, S. Yim, S. Choi and S. Choi, "Gesture-recognizing hand-held interface with vibrotactile feedback for 3D interaction," *IEEE Trans. Consum. Electron.*, vol. 55, no. 3, pp. 1169-1177, 2009.

[3] S. S. Rautaray, and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1-54, 2015.

[4] D. W. Lee, J. M. Lim, J. Sunwoo, I. Y. Cho and C. H. Lee, "Actual remote control: a universal remote control using hand motions on a virtual menu," *IEEE Trans. Consum. Electron.*, vol. 55, no. 3, pp. 1439-1446, 2009.

[5] D. Lee and Y. Park, "Vision-based remote control system by motion detection and open finger counting," *IEEE Trans. Consum. Electron.*, vol. 55, no. 4, pp. 2308-2313, 2009.

[6] F. Erden and A. E. Çetin, "Hand gesture based remote control system using infrared sensors and a camera," *IEEE Trans. Consum. Electron.*, vol. 60, no. 4, pp. 675-680, 2014.

[7] S. Jeong, J. Jin, T. Song, K. Kwon and J. W. Jeon, "Single-camera dedicated television control system using gesture drawing," *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1129-1137, 2012

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142-158, 2016.

[9] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, Santiago, 2015, pp. 1440-1448.

[10] S. Ren, K. he, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol.PP, no.99, pp.1-1, 2016.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR*, Las Vegas, NV, 2016, pp. 779-788.

[12] P. Molchanov, S. Gupta, K. Kim, and J. Kautz "Hand gesture recognition with 3D convolutional neural networks," in *Proc. CVPR*, Boston, MA, 22015, pp. 1-7.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. ECCV*, Zurich, 2014, pp. 346-361.

[14] M. A. Nielsen (2015, January 1). *Neural Networks and Deep Learning*, (1st ed.) [Online]. Available: http://neuralnetworksanddeeplearning.com.

[15] S. J. Nowlan, and G. E. Hinton. "Simplifying neural networks by soft weight-sharing," *Neural computation,* vol. 4, no. 4, pp. 473-493, 1992.

[16] G. E. Hinton, Geoffrey, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. (2012, July) Improving neural networks by preventing co-adaptation of feature detectors. Cornell University Library, NY. [Online]. Available: https://arxiv.org/pdf/1207.0580.pdf.

[17] N. H. Dardas, and N. D. Georganas. "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Trans. on Instrum. Meas.*, vol. 60, no. 11, pp. 3592-3607, 2011.

[18] X. Liu, and F. Kikuo, "Hand gesture recognition using depth data," in *Proc. ICAFGR*, 2004, pp. 529-534.

[19] S. B. Wang, A. Quattonni, L. P. Morency, and D. Demirdjian, "Hidden conditional random fields for gesture recognition," in *Proc. ICCV*, 2006, pp. 1521-1527.

[20] P. Trindade, J. Lobo, and J. P. Barreto, "Hand gesture recognition using color and depth images enhanced with hand angular pose data," in *Proc. ICMFIIS*, Hamburg, 2012, pp. 71-76.

[21] A. I. Maqueda, C. R. del-Blanco, F. Jaureguizar, and N. García, "Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns," *Comp. Vis. Image Underst.*, vol 141, pp. 126-137, 2015.

[22] Y. Wang, and Y. Ruoyu, "Real-time hand posture recognition based on hand dominant line using Kinect," in *Proc. ICME*, San Jose, CA, 2013, pp. 1-4.

[23] S. J. Nowlan, and J. C. Platt, "A convolutional neural network hand tracker," in *Proc. ICNIP*, Cambridge, MA, 1995, pp. 901-908.

[24] P. Sermanet, K. Kavukcuoglu, S. Chintala, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. ICCV*, Portland, OR, 2013, pp. 3626-3633.

[25] H. A. Rowley, B. Shumeet, and K. Takeo "Neural network-based face detection," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23-38, 1998.

[26] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Resear.*, vol. 15, no. 1, pp. 1929-1958, 2015.

[27] Y. Jia, E. Shelhamer, J. Donahue, and S. Karayev. (2014, June) Caffe: Convolutional architecture for fast feature embedding," Cornell University Library, NY. [Online]. Available: https://arxiv.org/pdf/1408.5093.pdf.

[28] T. Mantecón, A.I. Maqueda, C.R. del-Blanco, F. Jaureguizar, and N. García. (2017, March). Image database for tiny hand gesture recognition. [Online]. Available: https://sites.google.com/view/handgesturedb/home.

[29] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Proc. NIPS*, Nevada, 2012, pp.1097-1105.

[30] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, San Diego, 2015, pp. 1-14.

**Peijun Bao** is a visiting student in the Image Processing Group of Universidad Politécnica de Madrid, Spain. Previously, he was an undergraduate student in Northwestern Politecnical University, Xi’an, China, majoring in computer science and technology. His professional interests include machine learning, computer vision and neural network.

**Ana I. Maqueda** received the Telecommunication Engineering degree (integrated BSc-MSc accredited by ABET) from the Universidad Politécnica de Madrid (UPM) in 2014. Since 2015 she has been a PhD student in the Image Processing Group of the UPM. Her research interests include computer vision, machine learning, and augmented reality.

**Carlos R. del-Blanco** received the Telecommunication Engineering and Ph.D. degrees in telecommunication from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2005 and 2011, respectively. Since 2005, he has been a member of the Image Processing Group, UPM. Since 2011, he has also been a member of the faculty of the ETS Ingenieros de Telecomunicación as an Assistant Professor of signal theory and communications at the Department of Signals, Systems, and Communications. His professional interests include signal and image processing, computer vision, pattern recognition, machine learning, and stochastic dynamic models.

**Narciso García** received Telecommunication Engineering degree and the Ph.D. degree in Telecommunication, both from the Universidad Politécnica de Madrid (UPM), in 1976 (Spanish National Graduation Award) and 1983 (Doctoral Graduation Award), respectively. Since 1977 he is a member of the faculty of the UPM, where he is currently Professor of Signal Theory and Communications. He leads the Image Processing Group of the UPM. He was Coordinator of the Spanish Evaluation Agency from 1990 to 1992 and evaluator, reviewer, and auditor of European programs since 1990. His professional and research interests are in the areas of digital image and video compression and of computer vision.