Hand Posture Recognition Using Convolutional Neural Network

Dennis Núñez Fernández[2] and Bogdan Kwolek[1]

[1] AGH University of Science and Technology, 30 Mickiewicza, 30-059 Krakow, Poland
http://home.agh.edu.pl/~bkw/contact.html
[2] National University of Engineering, Av. Túpac Amaru 210, 15109 Lima, Peru

Abstract. In this work we present a convolutional neural network-based algorithm for recognition of hand postures on images acquired by a single color camera. The hand is extracted in advance on the basis of skin color distribution. A neural network-based regressor is applied to locate the wrist. Finally, a convolutional neural network trained on 6000 manually labeled images representing ten classes is executed to recognize the hand posture in a sub-window determined on the basis of the wrist. We show that our model achieves high classification accuracy, including scenarios with different camera used in testing. We show that the convolutional network achieves better results on images pre-filtered by a Gabor filter.

Keywords: Gesture recognition, Biologically inspired computer vision, Gabor filter, Convolutional neural network

1 Introduction

Hand gesture recognition is one evident way to build user-friendly interfaces between machines and their users. In the near future, hand gesture 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 necessity for physical contact between man and machine. Gesture recognition on images from single camera is a difficult problem due to occlusions, differences in hand anatomy, variations of posture appearance, etc. In the last decade, several approaches to gesture recognition on color images were proposed [1].

In recent years, Convolutional Neural Networks (CNNs) have become the state-of-the-art for object recognition in computer vision [2]. Despite high potential of CNNs in object detection [3,4] and image segmentation [2], only few papers report promising results - a recent survey on hand gesture recognition [1] reports only one significant work [5]. Some obstacles to wider use of CNNs are high computational demands, lack of sufficiently large datasets, as well as lack of hand detectors suitable for CNN-based classifiers. In [6], a CNN has been used for classification of six hand gestures expressed by humans to control robots using colored gloves. In more recent work [7], a CNN has been implemented in Theano and executed on the Nao robot. In a recent work [8], a CNN has been trained on one million of exemplars. However, only a subset of data with 3361 manually labeled frames in 45 classes of sign language is publicly available.

2 Method Overview

The proposed system for hand posture recognition operates on color RGB images. It has been designed to run in real-time with respect to low power usage, including devices without a GPU support. In order to reduce computational demands, the hand region proposals are determined at low computational cost on the basis of skin color. Given the extracted hand, a neural network based regressor is executed to estimate the wrist position, see Fig. 1. The wrist location is used to extract a sub-image with a hand, which is then fed to a CNN.

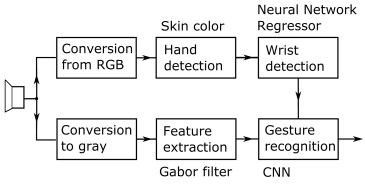

Fig. 1: Block diagram of our method for hand posture recognition.

The CNN operates on gray images of size 38×38. We trained two models for hand posture recognition: the first CNN was trained on gray images, whereas the second one was trained on gray images filtered by a Gabor filter. We show that the system achieves far better classification performance on images pre-filtered by a Gabor filter. We show that a CNN operating over pre-filtered images gives good results on both images from the camera that has been used to acquire images for training and images from a different camera, i.e. camera, whose images do not appear in the training repository.

The training of the CNN has been realized on a collection of 6000 images representing ten different hand postures. The network has been trained using Caffe [9]. The trained CNN models (Caffe Model Zoo), the training images as well as test images are available at: `http://home.agh.edu.pl/~bkw/code/ciarp2017`.

3 Gesture Dataset

The developed gesture set for interacting with computers is intuitive and easy to learn. It consists of ten gestures, see Fig. 2. The gestures were performed by ten persons of different nationalities. They were recorded using Kinect's RGB camera. The RGB images of size 640×480 were recorded in the `png` image format. On the basis of manually selected wrist position the sub-images containing the hands were determined. Afterwards, the background was automatically subtracted. Finally, all images were manually refined to eliminate remaining background pixels in the vicinity of the hands. Such manually cropped and refined sub-images of size 38×38 were stored in the image repository.

The whole dataset consists of 6000 gray images of size 38×38. Having on regard that such a dataset can be too little for building powerful classification models the hands were roughly aligned in such a way that characteristic hand features (e.g. the wrist) are approximately located at pre-defined positions in the

Fig. 2: Hand postures.

image. This means that in every class the wrists in all images are roughly located at the same position. Moreover, thanks to such an approach the recognition of gestures at satisfactory frame rates can be achieved with a simpler neural network and at a lower computational cost.

Until now, many hand datasets have been developed and made publicly available [10]. A total of 13 datasets with static hand gestures were available in the survey mentioned above at time of its publication. We found that only one set of data collections mentioned above has a sufficiently large number of examples per class, say 1000 images per class. With the discussed collection of benchmark data we selected dataset [11], albeit other dataset with large number of examples per class were also considered. As it turned out, the selected dataset does not fulfill our requirements well since it contains hands in front of background. Moreover, the hands are not properly aligned within classes. What is more, it contains the data that were shot by the same camera. Another issue is that the discussed dataset includes gestures, which are unlikely to have been designed for the control of the devices. Having the above on regard, we decided to record a hand posture dataset, which size is sufficient to perform deep learning. To the best of our knowledge, there is no other publicly available dataset of proper size, with appropriately aligned hands for learning of CNNs for real-time applications.

4 Hand Detection and Wrist Localization

At the beginning of this section, we describe how hands in RGB color images are detected. In the remainder of this section, we describe how the wrist is localized.

4.1 Hand Detection

Skin color is a powerful feature for fast hand detection. Basically, all skin color-based approaches try to learn a skin color distribution, and then use it to delineate the hands. In this work the hand has been extracted on the basis of statistical color models [12]. A model in RGB-H-CbCr color spaces has been prepared on the basis of a training set. Then, the hand probability image has been thresholded. Afterwards, after morphological closing, a connected components labeling has been executed to extract the gravity center of the blob, coordinates of the most top pixel as well as coordinates of the most left pixel of the hand region.

4.2 Wrist Localization

Modeling the statistical relation between the stochastic output y and input vector $\mathbf{x}$ is referred to as regression, which is typically expressed by an additive noise

model $y = g(\mathbf{x}) + e$, where e is a random error in y that cannot be expressed by the input. Taking into account the universal function approximation ability of neural networks, the regression model can be expressed as: $y = g(\mathbf{x}, \mathbf{w}) + \epsilon$, where ϵ is a random vector. Given the binary image representing the extracted hand, we extracted a feature vector $\mathbf{x}$ of size 38 elements. It consists of x, y-coordinates of gravity center, x, y-coordinates of the most left non-zero pixel, x, y-coordinates of the most top non-zero pixel, 16-bin histogram expressing the number of pixels in nonoverlapping column pairs, 16-bin histogram expressing the number of pixels in nonoverlapping row pairs. Two NN-based regression models were trained to model the relation between the feature vector $\mathbf{x}$ and wrist positions with respect to x, y coordinates. The weights $\mathbf{w}$ of a neural network with one hidden layer consisting of ten neurons were determined using standard backpropagation algorithm.

5 Hand Posture Recognition

At the beginning of this section, we describe how the input images are preprocessed by a Gabor filter. Then, we describe the convolutional neural network.

5.1 Gabor Filter

Hubel and Wiesel demonstrated in 1962 [13] the existence of simple cells in primary visual cortex (Nobel Prize, 1981). They discovered that its receptive field comprised sub-regions, which were layered over each other to cover the entire visual field. Gabor filter (GF) is excellent approximator of the receptive fields found in the mammalian primary visual cortex (V1). In the spatial domain, the 2D Gabor filters are Gaussian kernel functions modulated by a sinusoidal waves [14]. Their frequency and orientation representations are similar to those of the human visual system. The GF has a real and an imaginary component representing orthogonal directions. The real component can be expressed as:

$$g_{\lambda,\theta,\phi,\sigma,\gamma}(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \phi\right) \tag{1}$$

$$x' = x\cos\theta + y\sin\theta \qquad y' = -x\sin\theta + y\cos\theta \tag{2}$$

$$b = \log_2\left(\frac{\frac{\sigma}{\lambda}\pi + \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda}\pi - \sqrt{\frac{\ln 2}{2}}}\right), \quad \frac{\sigma}{\lambda} = \frac{1}{\pi}\sqrt{\frac{\ln 2}{2}} \cdot \frac{2^b + 1}{2^b - 1} \tag{3}$$

where λ is the wavelength of the cosine factor of the filter kernel, θ denotes the orientation of the normal to the parallel stripes of Gabor function, ϕ specifies the phase offset, γ is the spatial aspect ratio, that specifies the ellipticity of the support of the Gabor function, σ is the sigma/standard deviation of the Gaussian envelope, whereas b is related to the ratio $\frac{\sigma}{\lambda}$ and denotes the half-response spatial frequency bandwidth of Gabor filter.

5.2 Convolutional Neural Network

Unlike regular neural network, which only permits the input as vectors, convolutional neural networks allow 2 or 3-dimensional arrays at input layer. What makes convolutional neural networks distinct is that the weights are shared, that is, being different with respect to the position relative to the center pixel they are identical for different pixels in the image. Thus, it is straightforward to view a CNN as hierarchy of organized into layers a collection of local filters whose weights should be updated in a learning process. Every network layer acts as a detection filter for the presence of specific features or patterns present in the original data. The convolutions are usually followed by a non-linear operations after each layer since cascading linear convolutions would lead to a linear system. Besides, max-pooling is a mechanism that provides a form of translation invariance, which contributes towards the position independence. The CNNs are typically trained like standard neural networks using backpropagation.

In this work we utilized a convolutional neural network that is similar to the LeNet CNN [15]. The original model of LeNet relies on convolutional filters layers that are interlaced with non-linear activation functions, followed by spatial feature pooling operations, e.g. sub-sampling. It has only 60K learnable parameters out of 345 308 connections. Our convolutional neural network additionally uses ReLU activation functions, which were proposed in the Alexnet CNN [2]. The discussed work proposed also dropout regularization to selectively ignore single neurons during training, a way to avoid overfitting of the model.

Figure 3 depicts the architecture of our convolutional neural network. It consists of seven layers and takes a 38×38 pixel field as input.

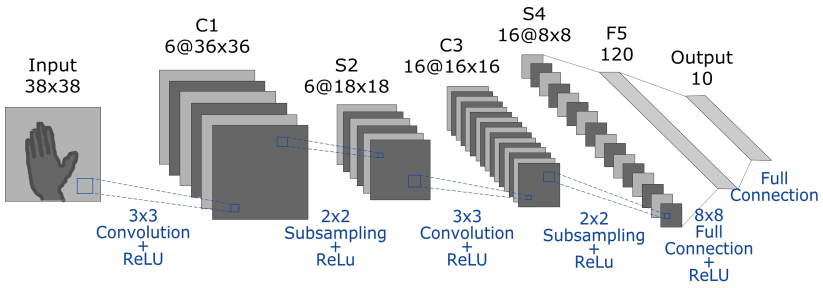

Fig. 3: Architecture of convolutional neural network.

6 Experimental Results

First, we trained CNNs on the hand posture dataset. In order to evaluate their generalization capability we additionally recorded images with the considered hand postures using a notebook camera. The camera that was utilized in image recording delivers images of size 640×480, i.e. the images have the same size as images acquired by Kinect RGB camera. Two volunteers of different nationalities, who did not attend in the dataset recordings performed 400 gestures.

Afterwards, we evaluated the error of wrist localization using the skin-color based algorithm for hand extraction and neural network based algorithm for wrist localization. The error was determined on 727 images (person #3) from the dataset and 400 images recorded by the notebook camera. On such an image set we manually determined the wrist positions and then used them to calculate the errors. Table 1 depicts the errors for each class together with the averages both for the Kinect and notebook camera (Dif. cam.). As we can notice, the average error for the different camera is slightly larger.

Table 1: Error of wrist localization [pix] for classes 1–10.

	1	2	3	4	5	6	7	8	9	10	avg.
Kinect	1.2	1.3	7.8	1.4	2.1	1.8	2.6	1.3	1.2	1.4	2.2
Dif. cam.	4.6	3.4	2.9	3.5	1.7	2.7	2.7	3.8	4.6	2.6	3.2

In the next stage, for each person, we divided the images into training and testing parts. About 15% of images from each class were selected for the tests, whereas the remaining images were used in the learning of CNNs. The images with gestures performed by person #3 were not included in the training data and they were used only in person-independent tests. This way we selected 6000 gray images of size 38×38 for training the CNNs. All CNNs were trained using Caffe [9] with the following parameters: batch size=64, momentum=0.9, base learning rate=0.001, gamma=0.1, step size=1000, max. iteration=5000.

The first convolutional neural network was trained on raw gray images. Table 2 depicts the recognition performance with respect to both automatically and manually determined wrist position on test images acquired by the notebook camera and Kinect camera, i.e. test part of the hand posture dataset. As we

Table 2: Performance measures using CNN on raw gray images.

	Accuracy	Precision	Recall	F1 score
Kinect man.	0.950	0.945	0.960	0.949
Dif. cam. man.	0.928	0.931	0.937	0.934
Kinect aut.	0.905	0.906	0.932	0.906
Dif. cam. aut.	0.783	0.788	0.832	0.785

can notice, on images acquired by Kinect the CNN achieves good classification performance. In the discussed performer-independent test the results are slightly worse for the wrist positions determined automatically. As we can observe, despite that the classification was done on images from a different camera, the CNN quite well recognizes the hand postures expressed by a different performer.

Table 3 depicts classification results that were obtained on images preprocessed by the Gabor filter. As we can observe, the results are far better in com-

parison to results presented in Table 2. The classification results achieved on images filtered by the Gabor filter are better for both the Kinect and notebook camera. In the person-independent test the CNNs achieves the classification accuracy equal to 97% if the images are taken by the same camera as in the training. If the classification is done on images acquired by different camera the classification accuracy is equal to 87%. For the discussed case the improvement with respect to classification on raw images is equal to 8.5%. The discussed results were achieved using the following parameters of the Gabor filter: $\lambda = 2$, $\theta = [0\ \pi/4\ \pi/2\ 3/4\pi]$, $\phi = [0\ \pi/2]$, $b = 1.8$, $\sigma = 0.75$, $\gamma = 0.5$.

Table 3: Performance measures using CNN with Gabor-based preprocessing.

	Accuracy	Precision	Recall	F1 score
Kinect man.	0.992	0.992	0.992	0.992
Dif. cam. man.	0.930	0.930	0.936	0.929
Kinect aut.	0.970	0.967	0.976	0.970
Dif. cam. aut.	0.868	0.868	0.882	0.866

Table 4 depicts classification accuracies that were achieved with respect to the considered hand postures. As we can notice, the CNNs achieves significantly worse results for class 6 when it operates on images taken by a different camera.

Table 4: Recognition accuracy [%] for wrist position determined automatically.

	1	2	3	4	5	6	7	8	9	10	avg.
Kinect	100	94	100	100	90	100	83	100	100	100	97
Dif. cam.	93	95	80	95	93	63	90	88	73	100	87

Our system has been designed to operate on humanoid Robot Nao as well as ARM processor-based mobile devices. Having on regard that they are not equipped with GPUs, which can considerably reduce the processing time of CNNs, we extract low-level features using techniques from biologically inspired computer vision, which are then further processed hierarchically, and finally recognized by moderate-size CNN. The recognition of hand posture on a single sub-image with the extracted hand and the estimated wrist position is about 5 ms. The presented software has been developed in C++, Python and Matlab.

7 Conclusions

There is still strong need for RGB-based hand posture recognition, particularly for mobile devices, and even in robotics since the cameras relying on structured light typically perform poorly under natural illumination. Despite huge number of approaches [1], only few papers demonstrated complete solutions, where

the images are taken by a RGB camera and both detection and classification algorithms are able to deliver promising results in real-time.

In this work we have presented CNN-based algorithm for hand posture recognition on images acquired by a single color camera. We considered ten static gestures for the operation of machines or smart devices. The CNNs have been trained on a repository of 6000 images of size 38×38. Two CNNs were trained both on raw images as well as images prefiltered by a Gabor filter. We demonstrated that CNN operating on prefiltered images has far better classification performance. We showed that our model allows for high classification accuracy even when the recognition is done on images taken by a different camera, i.e. a camera that has not been used in acquiring images for training of the CNNs.

Acknowledgment. This work was supported by Polish National Science Center (NCN) under a research grant 2014/15/B/ST6/02808.

References

1. Oyedotun, O., Khashman, A.: Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications (2016) 1–11
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS. (2012) 1097–1105
3. Kwolek, B.: Face detection using convolutional neural networks and Gabor filters. In: Int. Conf. Artificial Neural Networks, LNCS, vol. 3696, Springer (2005) 551–556
4. Arel, I., Rose, D., Karnowski, T.: Research frontier: Deep machine learning–a new frontier in artificial intelligence research. Comp. Intell. Mag. **5**(4) (2010) 13–18
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) (2014)
6. Nagi, J., Ducatelle, F., et al.: Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: IEEE ICSIP. (2011) 342–347
7. Barros, P., Magg, S., Weber, C., Wermter, S.: A multichannel convolutional neural network for hand posture recognition. In: 24th Int. Conf. on Artificial Neural Networks (ICANN), Cham, Springer (2014) 403–410
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 Conf. on Comp. Vision and Pattern Rec. (2016) 3793–3802
9. Jia, Y., Shelhamer, E., Donahue, J., et al.: Caffe: Convolutional architecture for fast feature embedding. In: ACM Int. Conf. on Multimedia. (2014) 675–678
10. Pisharady, P., Saerbeck, M.: Recent methods and databases in vision-based hand gesture recognition. Comput. Vis. Image Underst. **141** (2015) 152–165
11. Pisharady, P., Vadakkepat, P., Loh, A.: Attention based detection and recognition of hand postures against complex backgrounds. IJCV **101**(3) (2013) 403–419
12. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. Int. J. Comput. Vision **46**(1) (2002) 81–96
13. Hubel, D., Wiesel, T.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. **160**(1) (1962) 106–154
14. Petkov, N.: Biologically motivated computationally intensive approaches to image pattern recognition. Future Generation Computer Systems **11**(4-5) (1995) 451–465
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proc. of the IEEE. (1998) 2278–2324