In: Proceedings of the 24th International Conference on Artificial Neural Networks (ICANN 2014), LNCS 8681, pp. 403-410, Springer Heidelberg. Hamburg, DE, September 15-19, 2014.

A Multichannel Convolutional Neural Network for Hand Posture Recognition

Pablo Barros, Sven Magg, Cornelius Weber, and Stefan Wermter

University of Hamburg, Department of Computer Science,
Vogt-Koelln-Strasse 30, D-22527 Hamburg - Germany
{barros,magg,weber,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

Abstract. Natural communication between humans involves hand gestures, which has an impact on research in human-robot interaction. In a real-world scenario, understanding human gestures by a robot is hard due to several challenges like hand segmentation. To recognize hand postures this paper proposes a novel convolutional implementation. The model is able to recognize hand postures recorded by a robot camera in real-time, in a real-world application scenario. The proposed model was also evaluated with a benchmark database and showed better results than the ones reported in the benchmark paper.

Keywords: Hand Postures, Convolution Neural Networks, Deep Learning

1 Introduction

Gestures are recognized as crucial for human-human communication [12] and have inspired research for human-robot interaction [1] [4]. Hand gestures are widely used compared to other body parts [6], and thus are the main focus of most of the research in this field.

The most common approach for gesture recognition, as shown in the survey of Rautaray et al. [10] is the application of feature extraction techniques to represent postures. A popular feature extraction solution is to represent the hand by matching it to a template [4]. A problem with the template match approach is that a high variety of gestures executed by different kinds of people cannot be matched. Most of the feature extraction solutions need to segment the hand from the background of the image which can be done using a color scheme. Jmaa et al. [5] use an YCbCr color space model to separate the color information from the image luminance and obtain hand segmentation. This approach is not reliable if using a large variation in skin colors and luminance. Most of the applications using feature extraction use domain-based models and thus provide very specialized solutions [1].

Deep learning models for image classification have been studied in a vast number of experiments in the past few years [3]. Among deep learning techniques, Convolutional Neural Networks [7] have shown good results in the classification of static images [8]. The use of convolutional models focuses on how the human brain enhances and extracts features of an image in an implicit way using a set of local and global features.

This research describes a CNN, called Multichannel Convolutional Neural Network (MCNN), which allows the recognition of hand postures with implicit feature extraction. This novel architecture uses a cubic kernel concept and a multichannel flow of information, which allows it to recognize images even if they have a small size. The proposed architecture uses a similar concept of feature representation found in the research of Wallis et al. [14] and Wiskott et al. [15]. In their research, they create invariant responses of individual units to multiple instances of the same class. In the proposed model, this is possible with the implementation of a cubic kernel in the first convolutional layer. The research of [2] et al. uses convolutional layers stacked together to classify static images. In their research, several independent stacks of convolutional layers are put side-by-side and their results averaged to produce an improved classification rate. In their research, each stack receives the same image preprocessed with different algorithms. In our model, the channels are connected in the deep layers and trained together.

The model is evaluated in two different databases with static hand postures. One contains images taken by a robot in a home-like laboratory, simulating a real world scenario. There are four hand postures that are presented in different positions in each image. The other database is a benchmark database that contains ten different hand postures, three different backgrounds and has the hand always centered in the image.

2 Multichannel Convolutional Neural Network

A Convolutional Neural Network (CNN) is a set of pairs of convolution and max-pooling layers that enable the model to extract and enhance implicit features of an image. When stacked together, the first layers act like an edge enhancement and allow to extract local features which are passed to deeper layers which act like global feature extractors.

Each convolutional layer contains a set of feature maps, or filters, that extract features from a region of units using a convolution. Then an additive bias is applied and the result is passed through a sigmoid function. The value of a unit v_{nc}^{xy} in the position (x,y) at the nth feature map in the cth layer is given by

$$v_{nc}^{xy} = tanh\left(b_{cn} + \sum_{m}\sum_{h=0}^{H_{i-1}}\sum_{w=0}^{W_{i-1}} w_{ijm}^{hw} v_{(i-1)m}^{(x+h)(y+w)}\right), \tag{1}$$

where tanh is the hyperbolic tangent function, b_{cn} is the bias for the nth feature map of the cth layer, m indexes over the set of features maps in the (i-1) layer connected to the current layer c. In the equation, w_{nck} is the weight of the connection between the unit (h,w) within a region, or kernel, connected to the previous layer. H_i and W_i are the height and width of the kernel.

In the max-pooling layers, a region of the previous layer is connected to a unit in the current layer, reducing the dimension of the feature maps. For each layer, only the maximum value is passed. This enhances invariance to scale and distortions of the input [2]. The parameters of a CNN could be learned either by a supervised approach tuning the filters in a training database [3], or an unsupervised approach [9]. The proposed model uses the supervised approach.

2.1 Concept of a cubic kernel

In a CNN the convolution layers are applied on 2D feature maps to compute spatial features. To improve the feature enhancement the concept of a cubic kernel is applied. A cubic kernel is applied in a stack of images, simulating a 3D filter. The value of a unit (x,y,z) at the nth feature map in the cth layer is defined by

$$v_{nc}^{xyz} = tanh\left(b_{cn} + \sum_{m}\sum_{h=0}^{H_i-1}\sum_{w=0}^{W_i-1}\sum_{r=0}^{R_1-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right), \tag{2}$$

where z indexes the image in the image stack, R_i is the amount of pictures stacked together representing the new dimension of the kernel. In the proposed architecture the cubic kernel is applied to a stack of different images that belong to the same class, i.e. with the same hand posture in each image. This way the model can use the variance of images belonging to the same class to improve the tuning of the filters. To minimize the computational effort for the operation, a max operator is applied to the filter. The mean of the pixel intensities is calculated for each region and the pixel that presents the largest distance from the mean is used. This decreases the amount of connections and weights to be updated.

The same unit is connected to a different set of images but always to the same region in each image which makes the model create an average of the received regions. This gives the model the capacity to highlight pixel intensity variance for the class representation in the data, and allows the unit to learn faster than by using only one image. The tuning is improved by the fact that the connection is not shared between images, but each region in each image has its own weighted connection with the unit. This operation enhances the invariant responses within the same class, presenting the model with different pixel intensities in the same region of different images. The weights are tuned to learn this invariant response. This behavior can be found in the research of Wiskott et al. [15] and Wallis et al. [14]. The cubic kernel is applied only in the first convolutional layer, connected directly with the input images. For the recognition task, a set containing the same image is presented to the model instead of different images presented in the training task.

2.2 Multichannel implementation

To improve the tuning of the filters, a multichannel architecture is implemented in the proposed model. The idea behind this is to make use of existing knowledge,

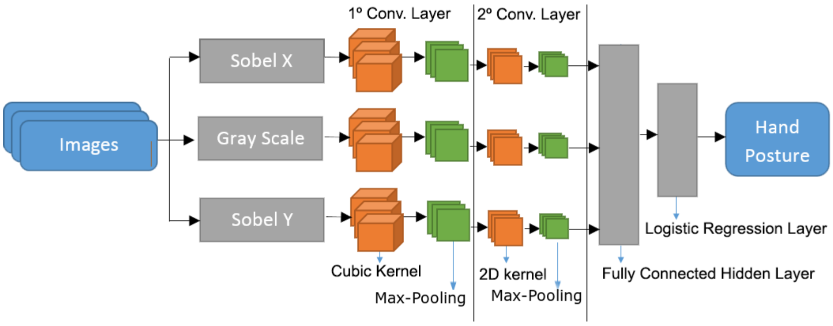
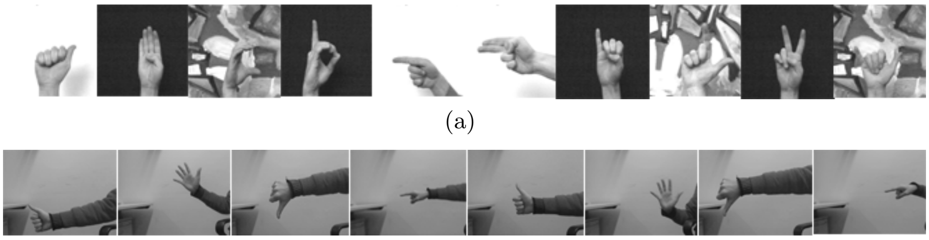


Fig. 1. Proposed architecture for a Multichannel Convolutional Neural Network using 3 channels. In this example, the MCNN has 2 layers and uses a logistic regression to recognize the gestures.

here represented in the edge enhancement by the Sobel operators, to diversify the input. Three channels are used: the first one receiving the raw image, the second and third one receiving the images resulting after applying a Sobel filter in both, horizontal and vertical directions, respectively. In the 3-channel CNN, each channel contains the same number of convolutional layers and the same parameters, but with independent weights. The three channels are connected with a fully-connected hidden layer that produces the output for a logistic regression classifier.

Each channel has its own weight update, but the final error is obtained with the output of the three layers, so that all the three layers act like a bias for each other. This produces a specialized filter tuning based on the edge enhancement of the Sobel filter in both directions. Figure 1 shows the architecture of the model which was used in the experiments.


(a)

(b)

Fig. 2. (a) Examples of hand postures on the JTD. (b) Examples of hand postures recorded with the NCD.

3 Experiments

To evaluate the model an experiment was performed using the Jochen Triesch Database (JTD) [13]. This database contains 10 hand postures, executed by 24 persons in front of 3 different backgrounds, a light, a dark and a complex one. All images are of size 128x128 and have the hand posture centered. Figure 2(a) shows examples of this database.

To evaluate the model in a realistic human-robot interaction scenario, a database using the camera of a small, 58cm tall, robot NAO[1] with four commands was recorded, called NAO Camera hand posture Database (NCD). Figure 2(b) shows example images. Each image has a resolution of 128x128 pixels, and the dataset has a large variance of executions, containing a total of 400-500 examples per hand posture. In each image, the hand was is present in different positions, not always in the centralized, and sometimes with occlusion of some fingers.

To evaluate how the proposed model works in different conditions and how the implementation of the cubic kernel and the three channels affect the final classification, a series of experimental setups was implemented. First, for each experiment, the images were presented in two ways: with the original size and with a reduced size of 28x28 pixels. Second, the experiments were realized with and without the cubic kernel. To evaluate the three channels, the experiments were compared with the utilization of a one-channel, two-channel and three-channel architecture. The network parameters, i.e. the number of convolutional layers, the dimension of the kernel and the max-pooling operation region were based on the research of [11]. The numbers of filters in each layer were found by evaluating the results for a range of numbers. Table 1 shows the parameters for each experimental setup. The parameters for the experiments without the cubic kernel were the same, but excluding the 3rd dimension of the kernel size.

4 Results and Discussion

Each experiment was executed 30 times and the mean of the F1-score for 1000 epochs was calculated. The database was divided into training, test and validation. For all the experiments, the selection was random and it used 60% of the database for training, 20% for validation and 20% for testing. All the experiments were implemented in Python using the library Theano[2] and were executed in a machine with an Intel Core 5i 2.67 Ghz processor, with 8GB of RAM.

The experiment results with the JTD and the cubic kernel, using all the backgrounds (light, dark and complex), are shown at Figure 3(a). The results show that the utilization of the specialized tuning with the three channels produces classification. The advantage of multiple channels is in particular striking for the small 28x28 images: when not using the Sobel operators, the architecture cannot extract or enhance any kind of efficient features, and it recognizes all the

[1] http://www.aldebaran-robotics.com/

[2] http://deeplearning.net/software/theano/

Table 1. Parameters for all the experiments evaluated in this paper.

Image	128x128			28x28		
		NCD	JTD		NCD	JTD
Layer 1	Filters	30	40	Filters	20	40
	Kernel Size	5x5x5	5x5x5	Kernel Size	5x5x5	5x5x5
	Subsampling Size	5x5	5x5	Subsampling Size	5x5	5x5
Layer 2	Filters	50	60	Filters	30	60
	Kernel Size	4x4	4x4	Kernel Size	2x2	2x2
	Subsampling Size	4x4	4x4	Subsampling Size	2x2	2x2
Layer 3	Filters	70	80	Filters	-	-
	Kernel Size	2x2	2x2	Kernel Size	-	-
	Subsampling Size	2x2	2x2	Subsampling Size	-	-

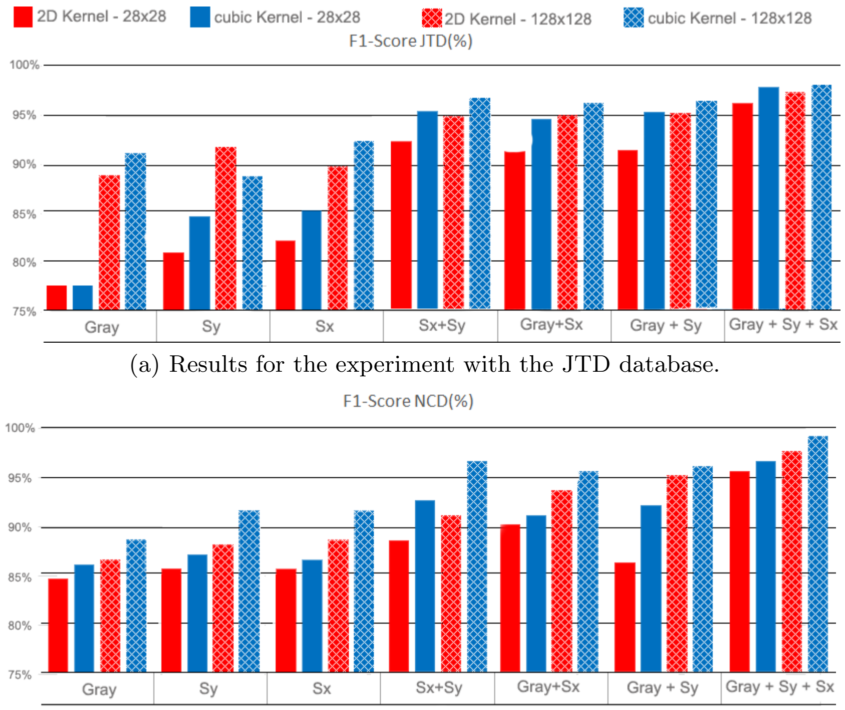
hand postures as posture 1. When we using only the Sobel filters, illustrated at Figure 3(a) by the columns Sy and Sx, the results show that the architecture still cannot extract an optimal set of features. Only when put together, the Sobel operator acts like a specialization for the original image, giving the model a bias for the edge enhancement, and produces better results.

There are no horizontal connections in the channels, making each channel independent. This makes the filters of each channel act only on the channel's own input. The weight update is guided by the three channels' results, but each filter must act differently for each kind of information it receives, otherwise the information specialization of each filter would be lost.

As expected, the recognition rates for the original image are larger than with the reduced image, but the difference is not so expressive when all the channels are combined. One point to be noted is the training and recognition time for the original image. In the original image, the mean time for recognition with the three-channel architecture is 0.125s. For the reduced image the recognition time is 0.0035s which is small enough to be used in real-time applications. The training time for the reduced image is also smaller, being 200.32s, and 1180.0s for the original image.

In the research of [13], they used an elastic graph matching to find the hand postures in the JTD. Taking together all images in all backgrounds, they obtained a 91% recognition rate. Our model obtained an F-1 score of 92% for the smaller images and 94% for the ones with original size. The model applied by [13] used a template-based match, which restricted the use of their solution in their own database. The nature of the MCNN allows it to be used in different databases without any specialized kind of preprocessing in the images.

The results on the NCD with and without the cubic kernel are shown in Figure 3(b). This experiment shows that with a large amount of data, the F1-scores obtained are very good. It shows also that the utilization of the cubic kernel and the three channels does not improve the results much more when there is a large amount of data, but still allows the model to recognize smaller images, which make the recognition and training tasks faster.



(a) Results for the experiment with the JTD database.

(b) Results for the experiment with the NCD database.

Fig. 3. F1-score for all experiments with all the combinations of architectures on (a) JTD and (b)NCD databases.

5 Conclusion

We developed a Multichannel Convolutional Neural Network for hand posture recognition. The model uses a cubic kernel to enhance the features for the classification and uses a multichannel architecture to specialize the tuning of the filters based on the Sobel operators. Each channel receives one kind of information that is used to represent efficient features for the presented database.

The proposed model was evaluated using two different databases: the Jochen Triesch hand posture database and a database recorded with the video camera of a NAO robot. The experiments in both databases show that the proposed model could recognize hand postures using a size-reduced image. The image reduction allowed the use of this deep learning model in recognition of hand postures in real time. For future research, it will be interesting to extend the proposed model to recognize dynamic gestures or multi-modal applications, with the addition of facial expression and audio data.

Acknowledgments. This research was partially supported by *Coordination for the Improvement of Higher Level -or Education- Personnel* (CAPES), Brazil (Grant number 5951-13-5).

References

1. S. Bilal, R. Akmeliawati, M. El Salami, and A. Shafie. Vision-based hand posture detection and recognition for sign language. In *4th International Conf. Mechatronics (ICOM)*, pages 1–6, May 2011.
2. D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012.
3. G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.
4. C.-H. Hu and S.-L. Wo. An efficient method of human behavior recognition in smart environments. In *International Conference on Computer Application and System Modeling (ICCASM)*, volume 12, pages 690–693, 2010.
5. A. Jmaa, W. Mahdi, Y. Jemaa, and A. Hamadou. Hand localization and fingers features extraction: Application to digit recognition in sign language. In *Intelligent Data Engineering and Automated Learning - IDEAL*, volume 5788, pages 151–159. 2009.
6. M. Karam. *PhD Thesis: A framework for research and design of gesture-based human-computer interactions.* PhD thesis, University of Southampton, October 2006.
7. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.
8. J. Nagi, F. Ducatelle, G. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. Gambardella. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 342–347, 2011.
9. M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 1–8, June 2007.
10. S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, pages 1–54, Nov. 2012.
11. P. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *International Conference on Document Analysis and Recognition*, pages 958–963, Aug 2003.
12. M. A. Singer and S. Goldin-Meadow. Children learn when their teachers' gestures and speech differ. *Psychological Science*, 16(2):85–89, 2005.
13. J. Triesch and C. V. D. Malsburg. Robust classification of hand postures against complex backgrounds. In *Interational Conference on Automatic Face and Gesture Recognition*, pages 170–175, 1996.
14. G. Wallis, E. Rolls, and P. Földiák. Learning invariant responses to the natural transformations of objects. In *International Joint Conference on Neural Networks*, pages 1087–1090, 1993.
15. L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.*, 14(4):715–770, Apr. 2002.