Sign Language Recognition using Convolutional Neural Networks

Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, Benjamin Schrauwen

Ghent University, ELIS, Belgium

Abstract. There is an undeniable communication problem between the Deaf community and the hearing majority. Innovations in automatic sign language recognition try to tear down this communication barrier. Our contribution considers a recognition system using the Microsoft Kinect, convolutional neural networks (CNNs) and GPU acceleration. Instead of constructing complex handcrafted features, CNNs are able to automate the process of feature construction. We are able to recognize 20 gestures from the Italian Sign Language (ISL). The predictive model is able to generalize on users and surroundings not occurring during training with a cross-validation accuracy of 91.7%. Our model achieves a mean Jaccard Index of 0.789 in the ChaLearn 2014 Looking at People gesture spotting competition, which ranks us fifth of the 17 teams.

Keywords: convolutional neural network, deep learning, gesture recognition, sign language recognition

1 Introduction

Very few people understand sign language. Moreover, contrary to popular belief, it is not an international language. Obviously, this further complicates communication between the Deaf community and the hearing majority. The alternative of written communication is cumbersome, because the Deaf community is generally less skilled in writing a spoken language [17]. Furthermore, this type of communication is impersonal and slow in face-to-face conversations. For example, when an accident occurs, it is often necessary to communicate quickly with the emergency physician where written communication is not always possible.

The purpose of this work is to contribute to the field of automatic sign language recognition. We focus on the recognition of the signs or gestures. There are two main steps in building an automated recognition system for human actions in spatio-temporal data [15]. The first step is to extract features from the frame sequences. This will result in a representation consisting of one or more feature vectors, also called descriptors. This representation will aid the computer to distinguish between the possible classes of actions. The second step is the classification of the action. A classifier will use these representations to discriminate between the different actions (or signs). In our work, the feature extraction is automated by using convolutional neural networks (CNNs). An artificial neural network (ANN) is used for classification.

2 Related Work

In our work, we build on the results of Roel Verschaeren [18]. He proposes a CNN model that recognizes a set of 50 different signs in the Flemish Sign Language with an error of 2.5%, using the Microsoft Kinect. Unfortunately, this work is limited in the sense that it considers only a single person in a fixed environment.

In [19] an American Language recognition system is presented with a vocabulary of 30 words. They constructed appearance-based representations and a hand tracking system to be classified with a hidden Markov model (HMM). An error rate of 10.91% is achieved on the RWTH-BOSTON-50 database.

The approach in [4] uses the Microsoft Kinect to extract appearance-based hand features and track the position in 2D and 3D. The classification results are obtained by comparing a hidden Markov model (HMM) approach with sequential pattern boosting (SP-boosting). This resulted in an accuracy of 99.9% on 20 different isolated gestures on their specifically constructed data set and 85.1% on a more realistic one with 40 gestures.

The Microsoft Kinect is also used in [2] that proposes a recognition system for 239 words of the Chinese Sign Language (CSL). Here, the 3D movement trajectory of the hands are used besides a language model to construct sentences. This trajectory is aligned and matched with a gallery of known trajectories. The top-1 and top-5 recognition rates are 83.51% and 96.32% respectively.

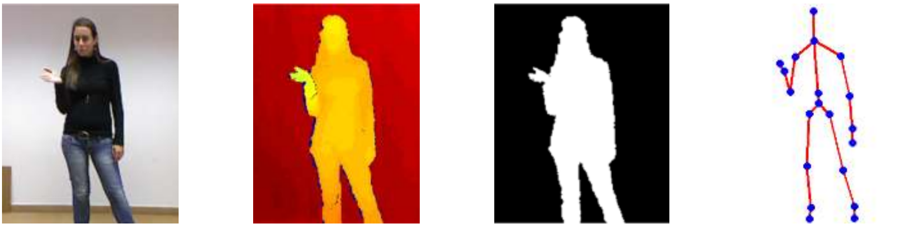

(a) RGB (b) Depth map (c) User index (d) Skeleton

Fig. 1. Data set for the CLAP14 gesture spotting challenge [5].

3 Methodology

3.1 Data

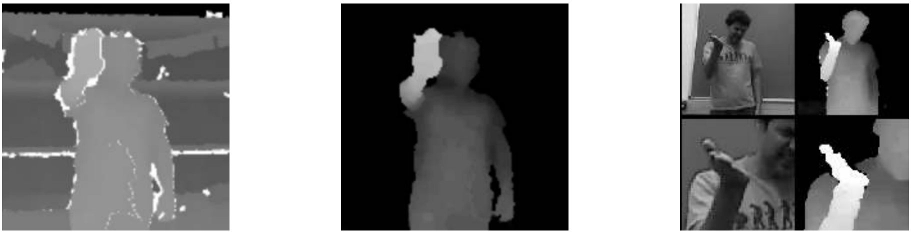
We use the data set from the *ChaLearn Looking at People 2014* [5] (CLAP14) challenge in this work. More specifically, *Track 3: Gesture Spotting*. This dataset consists of 20 different Italian gestures, performed by 27 users with variations in surroundings, clothing, lighting and gesture movement. The videos are recorded with a Microsoft Kinect. As a result, we have access to the depth map, user index (location of the user in the depth map) and the joint positions (Figure 1).

We use 6600 gestures in the development set of CLAP14 for our experiments: 4600 for the training set and 2000 for the validation set. The test set of CLAP14 is also considered as the test set for this work and consists of 3543 samples. The users and backgrounds in the validation set are *not* contained in the training set. The users and backgrounds in the test set *can* occur in the training and the validation set.

3.2 Preprocessing

Our first step in the preprocessing stage is cropping the highest hand and the upper body using the given joint information. We discovered that the highest hand is the most interesting. If both hands are used, they perform the same (mirrored) movement. If one hand is used, it is always the highest one. If the left hand is used, the videos are mirrored. This way, the model only needs to learn one side.

The preprocessing results in four video samples (hand and body with depth and gray-scale) of resolution 64x64x32 (32 frames of size 64x64). Furthermore, the noise in the depth maps is reduced with thresholding, background removal using the user index, and median filtering. The outcome is shown in Figure 2.



(a) Original (b) Noise reduction (c) 4 Input channels

Fig. 2. Preprocessing.

3.3 Convolutional Neural Network (CNN)

CNNs (based on [13]) are feature extraction models in deep learning that recently have proven to be to be very successful at image recognition [12], [3], [20], [7]. As of now, the models are in use by various industry leaders like Google, Facebook and Amazon. And recently, researchers at Google applied CNNs on video data [11].

CNNs are inspired by the visual cortex of the human brain. The artificial neurons in a CNN will connect to a local region of the visual field, called a receptive field. This is accomplished by performing discrete convolutions on the image with filter values as trainable weights. Multiple filters are applied for each channel, and together with the activation functions of the neurons, they form feature maps. This is followed by a pooling scheme, where only the interesting information of the feature maps are pooled together. These techniques are performed in multiple layers as shown in Figure 3.

3.4 Proposed Architecture

For the pooling method, we use max-pooling: only the maximum value in a local neighborhood of the feature map remains. To accommodate video data, the max-pooling is performed in three dimensions. However, using 2D convolutions resulted in a better validation accuracy than 3D convolutions.

The architecture of the model consists of two CNNs, one for extracting hand features and one for extracting upper body features. Each CNN is three layers deep. A classical ANN with one hidden layer provides classification after concatenating the outcomes of both CNNs. Also, local contrast normalization (LCN) as in [10] is applied in the first two layers and all artificial neurons are rectified linear units (ReLUs [14], [6]). An illustration of the architecture is depicted in Figure 3.

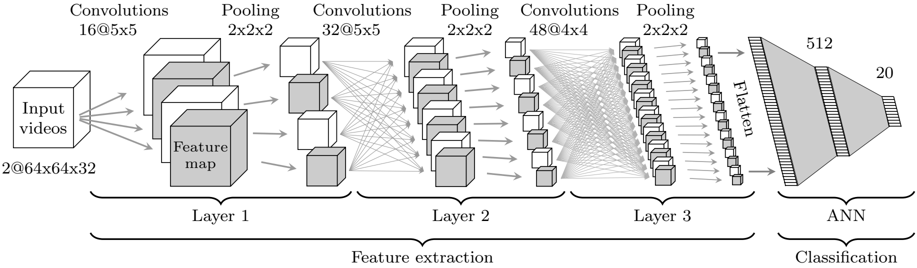


Fig. 3. The architecture of the convolutional neural network. This illustration only shows one of the two identical CNNs.

3.5 Generalization and Training

During training, dropout [9] and data augmentation are used as main approaches to reduce overfitting. The data augmentation is performed in real time on the CPU during the training phase whiles the model trains on the GPU as in [12]. This consists of zooming up to 10%, rotations up to (-)3°, spatial translations up to (-)5 pixels in the x and y direction, and temporal translations up to (-)4 frames.

We use Nesterov's accelerated gradient descent (NAG) [16] with a fixed momentum-coefficient of 0.9 and mini-batches of size 20. The learning rate is initialized at 0.003 with a 5% decrease after each epoch. The weights of the CNNs are randomly initialized with a normal distribution with $\mu = 0$ and $\sigma = 0.04$, and $\sigma = 0.02$ for the weights of the ANN. The biases of the CNNs are initialized at 0.2 and the biases of the ANN at 0.1.

Experiments are conducted on one machine with a hexa-core processor (Intel Core i7-3930K), 32GB SDRAM and a NVIDIA GeForce GTX 680 GPU with 4096MB of memory . The models are implemented using the Python libraries Theano [1], and PyLearn2 [8] for the fast implementation of 2D convolutions by Alex Krizhevsky [12].

3.6 Temporal Segmentation

The CLAP14 challenge consists of spotting gestures in video samples. Each video sample is an unedited recording of a user signing 10 to 20 gestures, including *noise* movements that are not part of the 20 Italian gestures. The goal of the temporal segmentation method is to predict the begin and end frames of every gesture in the video samples.

We use the sliding windows technique, where each possible interval of 32 frames is evaluated with the trained model (as previously described). Consecutive intervals with identical classes and sufficiently high classification probability (thresholding) are considered as a gesture segment. The validation set of CLAP14 is used to optimize the thresholding parameters. Furthermore, an extra class is added to the classifier to help identify video intervals without gesture.

Table 1. Validation results.

	Error rate (%)	Improvement (%)
Tanh units	18.90	
ReLU	14.40	**23.8**
+ dropout	11.90	17.4
+ LCN (first 2 layers)	10.30	13.4
+ data augmentation	**8.30**	19.4

4 Results

Our most notable experiments are the models with ReLUs, dropout, LCN and data augmentation. The validation results of these experiments are shown in Table 1. We observe a validation accuracy of 91.70% (8.30% error rate) for our best model. Furthermore, ReLUs prove to be very effective with an improvement of 23.8% with respect to tanh units.

The accuracy on the test set is 95.68% and we observe a 4.13% false positive rate, caused by the noise movements. Note that the test result is higher than the validation result, because the validation set doesn't contain users and backgrounds in the training set.

The final score for the CLAP14 competition is the mean Jaccard Index of each gesture and video sample. The Jaccard Index is a measure for overlapping frames between prediction and ground truth. The validation score of our best model is 0.789675 and the final score is 0.788804, which ranks us fifth of the 17 qualified teams.

5 Conclusion

This work shows that convolutional neural networks can be used to accurately recognize different signs of a sign language, with users and surroundings not included in the training set. This generalization capacity of CNNs in spatio-temporal data can contribute to the broader research field on automatic sign language recognition.

References

1. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy) (Jun 2010), oral Presentation
2. Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X., Zhou, M.: Sign Language Recognition and Translation with Kinect (2013), http://vipl.ict.ac.cn/sites/default/files/papers/files/2013_FG_xjchai_Sign Language Recognition and Translation with Kinect.pdf
3. Cireşan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 3642–3649. IEEE (2012)
4. Cooper, H., Ong, E.J., Pugeault, N., Bowden, R.: Sign language recognition using sub-units. The Journal of Machine Learning Research 13(1), 2205–2231 (2012)
5. Escalera, S., Bar, X., Gonzlez, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce, V., Escalante, H.J., Shotton, J., Guyon, I.: Chalearn looking at people challenge 2014: Dataset and results. In: ECCV workshop (2014)
6. Glorot, X., Bordes, A., Bengio, Y.: Deep Sparse Rectifier Networks. Proceedings of the 14th International Con- ference on Artificial Intelligence and Statistics 15, 315–323 (2011), http://eprints.pascal-network.org/archive/00008596/
7. Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv preprint arXiv:1312.6082 (2013)
8. Goodfellow, I.J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., Bengio, Y.: Pylearn2: a machine learning research library. arXiv preprint arXiv:1308.4214 (2013), http://arxiv.org/abs/1308.4214
9. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
10. Jarrett, K., Kavukcuoglu, K.: What is the best multi-stage architecture for object recognition? Computer Vision, 2009 IEEE 12th International Conference on pp. 2146–2153 (2009), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5459469
11. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
12. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Advances in Neural Information pp. 1–9 (2012), http://books.nips.cc/papers/files/nips25/NIPS2012_0534.pdf
13. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11) (1998)
14. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). pp. 807–814 (2010)
15. Poppe, R.: A survey on vision-based human action recognition. Image and vision computing 28(6), 976–990 (2010)
16. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13). pp. 1139–1147 (2013)
17. Van Herreweghe, M.: Prelinguaal dove jongeren en nederlands: een syntactisch onderzoek. Universiteit Gent. Faculteit Letteren en Wijsbegeerte (1996)

18. Verschaeren, R.: Automatische herkenning van gebaren met de microsoft kinect (2012)
19. Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision based features. Pattern Recognition Letters 32(4), 572–577 (2011)
20. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional neural networks. arXiv preprint arXiv:1311.2901 (2013)