Hand Gesture Recognition using Deep Convolutional Neural Networks

Conference Paper · September 2016

CITATIONS
4

READS
6,595

4 authors:

Gjorgji Strezoski
University of Amsterdam
23 PUBLICATIONS **400** CITATIONS
SEE PROFILE

Dario Stojanovski
Ludwig-Maximilians-University of Munich
22 PUBLICATIONS **233** CITATIONS
SEE PROFILE

Ivica Dimitrovski
Ss. Cyril and Methodius University in Skopje
61 PUBLICATIONS **704** CITATIONS
SEE PROFILE

Gjorgji Madjarov
Ss. Cyril and Methodius University in Skopje
41 PUBLICATIONS **1,012** CITATIONS
SEE PROFILE

Hand Gesture Recognition using Deep Convolutional Neural Networks

Gjorgji Strezoski, Dario Stojanovski, Ivica Dimitrovski, and Gjorgji Madjarov

Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University,
Rugjer Boshkovikj 16, 1000 Skopje, Macedonia
gjorgji.strezoski@finki.ukim.mk, dario.stojanovski@finki.ukim.mk,
ivica.dimitrovski@finki.ukim.mk, gjorgji.madjarov@finki.ukim.mk

Abstract. Hand gesture recognition is the process of recognizing meaningful expressions of form and motion by a human involving only the hands. There are plenty of applications where hand gesture recognition can be applied for improving control, accessibility, communication and learning. In the work presented in this paper we conducted experiments with different types of convolutional neural networks, including our own proprietary model. The performance of each model was evaluated on the Marcel dataset providing relevant insight as to how different architectures influence performance. Best results were obtained using the GoogLeNet approach featuring the Inception architecture, followed by our proprietary model and the VGG model.

Keywords: gesture recognition, computer vision, convolutional neural networks, deep learning, Inception architecture, GoogLeNet

1 Introduction

Hand gestures provide a separate complementary modality to speech for expressing ones ideas. Information associated with hand gestures in a conversation is degree, discourse structure, spatial and temporal structure. The approaches present can be mainly divided into Data-Glove based and Vision based approaches [1]. These two approaches are fundamentally different due to the different nature of the sensory data collected. The Data-Glove based approach collects data from sensors attached to a glove mounted on the hand of the user. Using this methodology only necessary information is gathered, which minimizes the need of data preprocessing and reduces the amount of junk data. Nevertheless, using a Data-Glove in real life scenarios is often infeasible and can present different issues like connectivity, sensor sensitivity and many other hardware related problems [2].

Vision based approaches on the other hand offer the convenience of hardware simplicity - they only require a camera or some sort fo scanner. This type of approach complements biological human vision by artificially describing the visual field. While this type of approach is way cheaper than its Data-Glove counterpart, it produces a large body fo data that need to be carefully processed in order to get only the necessary information. Having in mind that to tackle

this problem the recognition system needs to be insensitive to lighting conditions, background invariant and also subject and camera independent [3]. Also a challenging part of the hand gesture recognition problem is the fact that these systems need to provide real-time interaction. While this does not affect the model training directly it implies that later classification needs to be conducted in a manner of milliseconds.

Given the constraints presented by the nature of hand gesture recognition, a generally invariant approach is required which will retain consistent performance in various conditions. In recent years deep learning has stepped up the game when it comes to computer vision problems. Deep learning approaches have shown to be superior in various computer vision challenges on multiple topics. This spike in the performance of these models is partially due to the recent advances in GPU design and architectures. GPUs are parallel in nature and are especially well adjusted for training these types of models. While there is a vast variety of deep architectures, research has shown that Convolution Neural Networks (CNNs) are most applicable to computer vision problems. This compatibility rests on the biological similarity of convolutional neural networks with the vision part of the human brain [4]. Having said that, humans have the most sophisticated vision system, which similarly to convolutional neural networks consists of hierarchically distributed layers of neurons which act as processing units. Parameter sharing between neurons from different levels in the structure yield different connection patterns with different connection weights, which in turn concludes the process with classification.

Since these types of architectures have gained popularity during the past few years, the industry leaders like Google, Nvidia, Microsoft, Deep Mind, IBM, Clarifai and others have developed their own architectures designed to tackle diverse problems. Most of these architectures are available for personal and academic purposes under open licenses ergo researchers and professionals alike can modify the code, adjust the model and fine-tune the existing parameters. There is also a vast academic community which continuously pushes performance limits of deep models. Berkeley developed Caffe which is one of the best performing deep learning framework and Oxfords Visual Geometry Group introduced state of the art performance with a weakly supervised deep detection architecture. Similarly to these advances Microsoft released its deep learning flagship CNTK, Google released TensorFlow and Nvidia released the cuDNN framework which optimizes GPU operations for maximum performance.

Having given a brief introduction into the field, our research contributions to this area are three fold:

- We evaluate several plain and pre-trained convolutional neural networks on different datasets and compare their performance.
- We train a robust deep model for hand gesture recognition with high accuracy rate.
- We report good performance in a temporal manner with just 2ms classification time on the fully trained model, making it a real-time functional model.

The rest of the paper is organized as follows. Section 2 outlines current approaches on hand gesture recognition, with emphasis deep learning methods and state of the art performance. Section 3 presents the details of the experimental scheme and an overview of the pre-trained and plain models used in our work. We present and elaborate on the performance achieved using every approach and provide insight on the findings of our research in Section 4. Finally, we conclude our work and discuss future development in Section 5.

2 Related work

Hand gestures are a fundamental part in human-human communication [2]. The efficiency of information transfer using this technique of communication is remarkable, therefore it has sparked ideas for utilization in the area of human-computer interaction. For this to be possible the computer needs to recognize the gesture shown to it by the person controlling it. That is the sole process of hand gesture recognition. In a classical manner, the most common approach to solving these types of problems is applying feature extraction techniques. A particular technique is matching the image of the hand a predefined template [6]. Template matching has shown to be ineffective due to the high variety of environments, hand forms and variations of different gestures. Other classical approaches featuring different feature extractors have the flaw of not being flexible enough to changing datasets and alternating conditions. In these cases the robustness and invariance of the approaches with deep convolutional neural networks makes them ideal candidates for these types of problems.

As we mentioned before, deep learning methods have been used to solve a diverse field of computer vision problems in recent years. When it comes to problems that are representable via images, the parallel nature of convolutional neural networks allows them to elegantly apply to the matrix representation of the data. Additionally, multi-column deep CNNs that employ multiple parallel networks have been shown to improve recognition rates of single networks by 30-80% for various image classification tasks [7]. Neverova et al. [8] successfully combined RGBD data from the hand region with upper-body skeletal motion data using convolutional neural networks (CNNs) for recognizing 20 Italian sign language gestures. However, their technique was intended for gestures performed indoors only. Pablo Barros et al. [9] designed a Multichannel Convolutional Neural Network (MCNN) which allows hand gesture recognition with implicit feature extraction in the architecture itself. They report state of the art results on two dataset containing images of static hand gestures. The first dataset was generated using a robot in laboratory conditions, mimicking real world scenarios with four types of hand gestures. As a secondary dataset, they evaluated their system on data containing ten different hand gestures made in real, uncontrolled environments.

Ohn-Bar and Trivedi evaluated various handcrafted spatio-temporal features and classifiers for in-car hand-gesture recognition with RGBD data [3]. They reported the best performance with a combination of histogram of gradient (HOG)

features and an SVM classifier. Molchanov et al. fused information of hand gestures from depth, color and radar sensors and jointly trained a convolutional neural network with it. They demonstrated successful classification results for varying lighting conditions and environments [7]. In turn, the before mentioned efforts provided the necessary background for conducting our experiments and motivated our work.

3 Experimental design

Recent advances in the design of models with deep architectures, especially convolutional networks have paved the way for a vast number of different CNN architectures designed to handle all sorts of data. Following the work in [7] [8] [9] [10] [11] [14] we decided to test the best performing models in some of the most challenging visual classification tasks like the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), on hand gesture recognition. Furthermore, we propose our own CNN designed with robustness and efficiency in mind.

3.1 Dataset

For the purpose of training and testing our model we used the Marcel dataset which consists of 6 hand signs (A, B, C, FIVE, POINT, V) performed by 24 persons on three different types backgrounds. Different people and background were used in order to increase diversity and information contained within the dataset. In terms of background, the images in the Marcel dataset were recorded in front of an uniform light background, uniform dark background and a complex background [12]. Because of the different people included in the creation of this dataset there are also variabilities in hand shape and sizes. This dataset results in a total of 4937 train images and 675 test images. For the testing and validation of the different models performance we used five fold cross-validation. The distribution of images in each of the classes in both the training and testing set are shown in Table 1.

Table 1: Number of images in each class

	Train	Test	Total
A	1331	99	1430
B	489	104	593
C	573	116	689
FIVE	655	138	793
POINT	1396	121	1517
V	436	97	533

3.2 Data Augmentation

Because the deep architectures that we trained in our experiments require a large mass of data to train properly, we used data augmentation on the images in the dataset. This was done in order to gain quantity while still introducing some novelty in terms of information to our dataset. Our augmentation consists of horizontal mirroring of every image in the training set, effectively doubling the size of the dataset [13]. Horizontal mirroring data augmentation is labeled label-safe in this type of images. Additionally we trained our models using a gray-scale representation, thus removing the color factor. Samples of this dataset on a light plain background are illustrated in Figure 1

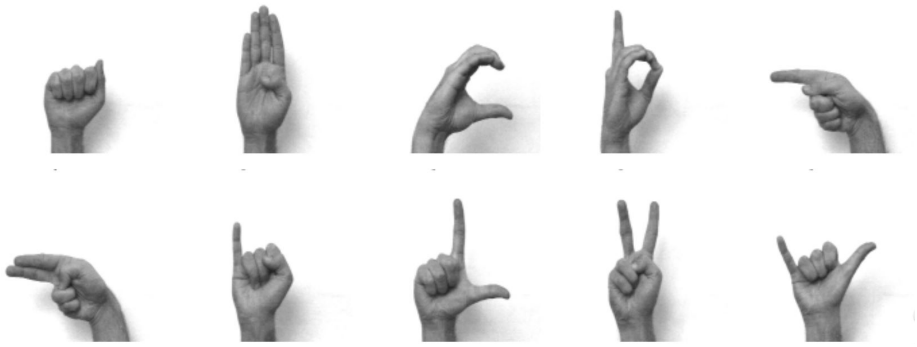

Fig. 1: Samples of the Marcel dataset on plain background

3.3 GoogLeNet

GoogLeNet is a deep convolutional neural network designed by Google featuring their popular Inception architecture. This architecture not only allows for approximation of a optimal local sparse structure by readily available dense components but also reduces data dimensionality [10] wherever the computational requirements would increase rapidly. GoogLeNet is the particular incarnation of the Inception architecture that had the lowest error rate in the ILSVRC 2014 challenge.

We trained this model on the Marcel training dataset for 30 epochs with a batch size of 16 images and a 20% step degradation function. The initial learning rate was setup to 0.001 and because of the mixed presence of different backgrounds (complex, plain light, plain dark). Also we subtracted the total RGB mean of the complete dataset before data iteration. Additionally all of the images were maximized to a size of 256x256px to fit the receptive field of the CNN. Whenever the original proportions of the input image could not be maintained, cropping was introduced to the central region of the image. The total training time of this model was 2 hours on a NVidia GTX 980 Ti.

3.4 AlexNet

Following its success on the ILSVRC-2010 challenge and relatively simple structure with just 5 convolutional layers and 3 fully connected layers, AlexNet provides the simplicity and efficiency of a shallow model combined with the predictive performance of a deep model [11].

We trained this model for 20 epochs using an initial learning rate of 0.001 and a batch size of 16 images. In order to encourage faster learning we applied an exponential learning rate degradation function with a gamma factor od 0.02. In this case we also subtracted the mean file from the input images in both the training and testing phase. The total training time of this model was 1 hour on a NVidia GTX 980 Ti.

3.5 LeNet

The LeNet model is specifically designed for handwritten and machine printed character recognition. Having in mind the similarity of the characters with hand gesture contours, whether printed or written, given the suitable preprocessing this model should perform well. This model features 7 layers (not counting the input layer) and a receptive field of 28x28px. This receptive field yields the need of a region-of-interest (ROI) selector or a smart cropping mechanism, in order to fit the images into the input space of the model. After cropping the images to their central section where the gesture is usually contained using PIL in Python, a resize function scaled the images to 28x28 pixels.

For this experiment we trained the LeNet model for 35 epochs with a initial learning rate od 0.01 and a step degradataion function of 25% step frequency.

3.6 VGG Net

The Visual Geometry Group model is described as a very deep convolutional neural network [14] that has a fixed input size of 224x224px RGB image. As a preprocessing step in the training process of this network we subtracted the mean RGB value, computed on the training set, from the input images. This model features a small receptive field and convolutional filters with 3x3px dimensions. The stack of convolutional layers also contain spatial pooling layers with a 2x2px window, that passes the data with stride 2 [14]. After the convolutional stack there is a series of three fully connected layers. The last fully connected layer performs the classification over the 6 classes.

We trained this model for 35 epochs with a batch size set to 128. Because this network is deeper than most other architectures it takes less epoch to converge, so during training we noticed convergence around the 17 epoch. Learning started with a base learning rate of 0.001, which degraded with a step degradation function 33%, 3 times on a regular interval.

3.7 Custom model

Our custom model was originally designed for pixel based segmentation of images. Since the process of segmentation essentially rests on pure classification, small corrections to the kernel and filter sizes of the architecture allowed for this architecture to achieve relatively good performance in this task. This model has 13 layers that contain 5 convolutional and 5 pooling layers [13]. Before the softmax classifier there is a fully connected layer aggregating the convolved features generated to this point. Finally the input layer creates a 194x194px receptive field.

Table 2: Layer configuration in Custom model

	Type	Units	Kernel
0	input	194x194	N/A
1	convolutional	192x192	4x4
2	max pooling	96x96	2x2
3	convolutional	92x92	4x4
4	max pooling	46x46	2x2
5	convolutional	42x42	5x5
6	max pooling	21x21	2x2
7	convolutional	18x18	4x4
8	max pooling	9x9	2x2
9	convolutional	6x6	5x5
10	max pooling	3x3	2x2
11	fully connected	600	1
12	softmax	6	1

This model was implemented in Berkeley's Caffe framework using the Python wrapper. Each of the neurons contained in this network relies on the Rectified Linear Unit activation firs introduced in [11]. In the custom model classification is performed using a Softmax classifier with 6 output neurons (one for each class). Table 2 shows the models configuration layer by layer with unit numbers and kernel sizes.

As with the previous models, this model was trained no more than 25 epochs in its best run. For increased performance (and reduced speed) we started training this model with an initial learning rate of 0.002 and degraded this rate using a 20% step degradation function.

4 Results and Discussion

In this particular research, it was important to evaluate performance in both accuracy and operation timing due to the potential of applying this types of models in a real-time control scenario. In terms of accuracy, the GoogLeNet model performed best with a Top-1 classification accuracy of 78.22% and Top-3 classification accuracy of 90.41%. While it is the best model in term of accuracy, because of its depth and complexity, classification and training times are the longest with 4 minutes per epoch in a training setting and 2.8ms propagation time of a test image from the input layer to the end of the network.

Table 3: Accuracy scores for each models best run

	Top-1	Top-3
GoogLeNet	78.22%	90.41%
AlexNet	42.18%	60.9%
Custom model	64.17%	84.32%
LeNet	28%	47.19%
VGG model	64.19%	83.33%

The VGG model and our custom model have similar accuracies in both the Top-1 and Top-3 categories, with 64.19% and 83.33% for the VGG model and 64.17% and 84.32% for our custom model respectfully. The AlexNet and LeNet models performed significantly worse than the previous three models as shown in Table 3.

Table 4: Average classification and training epoch duration on a GPU

	Training epoch (min)	Classification per image (ms)
GoogLeNet	4	2.8
AlexNet	2	1
Custom model	2.5	0.5
LeNet	0.4	0.6
VGG model	5	2

When it comes to classification and training duration, as expected the most simple and shallow architectures provide the best timings in both areas. The

LeNet model performed best in terms of timing with 0.4 minutes per epoch in the training setting and 0.6 milliseconds for predicting the class of a single image. Nevertheless, its good performance on the timing evaluation, the bad classification accuracy makes this model unsuitable for any kind of real time control. High error rate in classification would make the system unreliable. On the other hand the GoogLeNet model was the slowest of all tested models with 4 minutes per training epoch and almost three milliseconds per classification of a single image. Our custom model and the VGG model had the best ratio of classification accuracy and timing making them most suitable for real time use. The GoogLeNet model would also perform well in this type of setting but it would require more expensive hardware (GPU) and plenty of optimizations to gain in responsiveness. Responsiveness is a major concern in systems control and human-robot interaction.

5 Conclusion and Future Work

Regarding future work in this direction, we are exploring methodologies for improving our own model and its predictive power. Additionally, because of the short classification process duration, application of this model in a real-time setting is part of our future work as well. This is possible with XBox Kinect sensor technology, providing stable and consistent data feed to the model. Using the XBox Kinect sensor we would also be able to generate our own datasets for retraining and improving classification rates.

An interesting future approach would be the development of a continuous training of the model so that with each correct classification we would adjust the weights and activations of network it self. Basically that would enable the model to get progressively better with time, without the need of a separate training phase.

6 Acknowledgments

We would like to acknowledge the support of the European Commission through the project MAESTRA Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944).

References

1. Shastry, K.R., Ravindran, M., Srikanth, M., Lakshmikhanth, N., et al.: Survey on various gesture recognition techniques for interfacing machines based on ambient intelligence. arXiv preprint arXiv:1012.0084 (2010)
2. Singer, M.A., Goldin-Meadow, S.: Children learn when their teacher's gestures and speech differ. Psychological Science **16**(2) (2005) 85–89
3. Ohn-Bar, E., Trivedi, M.M.: Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. Intelligent Transportation Systems, IEEE Transactions on **15**(6) (2014) 2368–2377

4. Strezoski, G., Stojanovski, D., Dimitrovski, I., Madjarov, G.: Content based image retrieval for large medical image corpus. In: Hybrid Artificial Intelligent Systems. Springer (2015) 714–725
5. Stojanovski, D., Strezoski, G., Madjarov, G., Dimitrovski, I.: Emotion identification in fifa world cup tweets using convolutional neural network. In: Innovations in Information Technology (IIT), 2015 11th International Conference on, IEEE (2015) 52–57
6. Bilal, S., Akmeliawati, R., El Salami, M.J., Shafie, A.A.: Vision-based hand posture detection and recognition for sign languagea study. In: Mechatronics (ICOM), 2011 4th International Conference On, IEEE (2011) 1–6
7. Molchanov, P., Gupta, S., Kim, K., Kautz, J.: Hand gesture recognition with 3d convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2015) 1–7
8. Neverova, N., Wolf, C., Taylor, G.W., Nebout, F.: Multi-scale deep learning for gesture detection and localization. In: Computer Vision-ECCV 2014 Workshops, Springer (2014) 474–490
9. Barros, P., Magg, S., Weber, C., Wermter, S.: A multichannel convolutional neural network for hand posture recognition. In: Artificial Neural Networks and Machine Learning–ICANN 2014. Springer (2014) 403–410
10. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1–9
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
12. Marcel, S., Bernier, O., Viallet, J.E., Collobert, D.: Hand gesture recognition using input-output hidden markov models. In: fg, IEEE (2000) 456
13. Strezoski, G., Stojanovski, D., Dimitrovski, I., Madjarov, G.: Deep learning and support vector machine for effective plant identification
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)