GaborNet

# Abstract

This research on deep convolutional neural networks proposes a modified architecture that focuses on improving convergence and reducing training complexity. The filters in the first layer of network are constrained to fit the Gabor function. The parameters of Gabor functions are learnable and updated by standard backpropagation techniques. The proposed architecture was tested on several datasets and outperformed the common convolutional networks.

# Introduction

Convolutional neural networks (CNNs) have various applications, especially in computer vision. They have attracted significant attention due to its ability to be trained end-to-end and capability of learning outstanding feature representations from raw image data. In 2015 CNN showed human-like performance on ImageNet dataset [1]. Unlike classic computer vision methods, neural network is a data driven algorithm, that learns robust representations from data, but usually at the cost of training excessive number of parameters (or weights). Additionally, convergence of neural network depends on parameters initialization. Usually the weights are initialized by uniform or normal distribution. But it causes convergence problem and makes difficult to train very deep CNN.

Glorot and Bengio (2010) [2] proposed a formula, that estimates standard deviation for a CNN layer based on layer input and output size. In [1] new method of initialization, known as He initialization, was introduced, which also suggests using normal or uniform distribution with parameters based on neural network topology and type of activation functions. However, these approaches focused mainly on dealing with training deep CNN. Hence additional constraints and initialization methods should be applied to convolutional neural networks’ weights to overcome slow convergence speed. Specifically, this could be achieved by generating weights of neural network by family of parameterized filters.

Gabor filters [3] are widely used in computer vision. They are based on sinusoidal plane wave with particular frequency and orientation, which allows them to extract spatial frequency structures from images [4]. Additionally, these filters proved to be appropriate for texture representation and face detection tasks [5]. A few works have explored Gabor filters for CNN. In [6] Gabor filters were used as preprocessing tool to generate Gabor features then using it as an input to a CNN, in [7] first or second layer of CNN was set as a constant Gabor filter bank, thus reducing number of trainable parameters of the network, and in [8] Convolutional Gabor orientation Filters were introduced, a special structure that modulates convolutional layers with learnable parameters by non-learnable Gabor filter bank. However, they do not integrate Gabor filters into backpropagation algorithm.

In this paper we propose using Gabor Layer as a first layer in deep convolutional network. Gabor Layer is a convolutional layer, which filters are constrained to fit Gabor functions (Gabor filters). The parameters of Gabor functions are initialized from filter bank, proposed in [9], and during training process the filter parameters are updated by standard backpropagation algorithm. This approach aims to increase robustness of learned feature representations and to reduce training complexity of neural networks. Gabor Layer is implemented on basic elements of CNNs and can be easily integrated into any deep CNN architecture.

# Theory

## CNN

Convolutional neural networks usually consist of convolutional layers and fully connected layers, with nonlinear activation function applied at the end of each layer. Pooling and dropout layers may be included in the architecture to avoid overfitting. Convolutional layers reduce memory usage and increase performance by using the same filters for each pixel of the image. The standard architecture of CNN is shown in Fig.1.

The training process of CNN usually is carried out by backpropagation algorithm [10] with active dropout layers. The evaluation process is much simpler, because it requires only forward propagation, dropout layers is inactive during evaluation process.

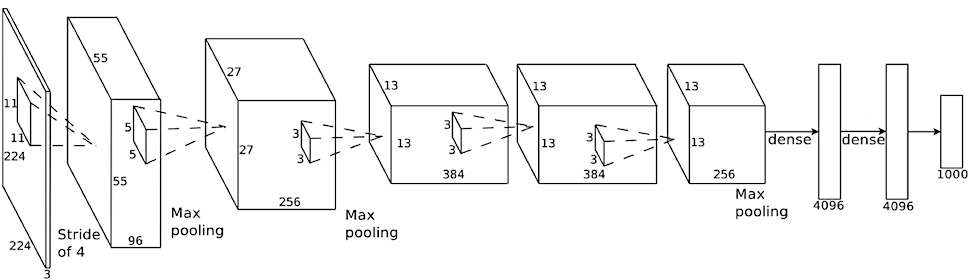


Fig 1 CNN structure (Alexnet) from http://ml4a.github.io/

## Gabor function

Gabor function is a complex sinusoid modulated by Gaussian i.e.

However, in this work we used real values of Gabor function i.e.

Gabor filters proved to be an efficient tool for extracting spatially localized spectral features, which are used in various pattern analysis applications. In [11] was shown that deep CNN trained on real-life images tends to learn first convolutional layers contain mostly Gabor-like filters. Filters of first and second layers of AlexNet is shown in Fig.2. This corroborates an idea of using Gabor filters in first layer of CNN.

To initialize Gabor filters in Gabor Layer, approach proposed in [12] was used. Frequencies and orientations of the Gabor filters are obtained by the following equations:

The is set by , which allows to define relationship between and . The is set by uniform distribution U(0, ). During the training process these parameters are learned by backpropagation algorithms.

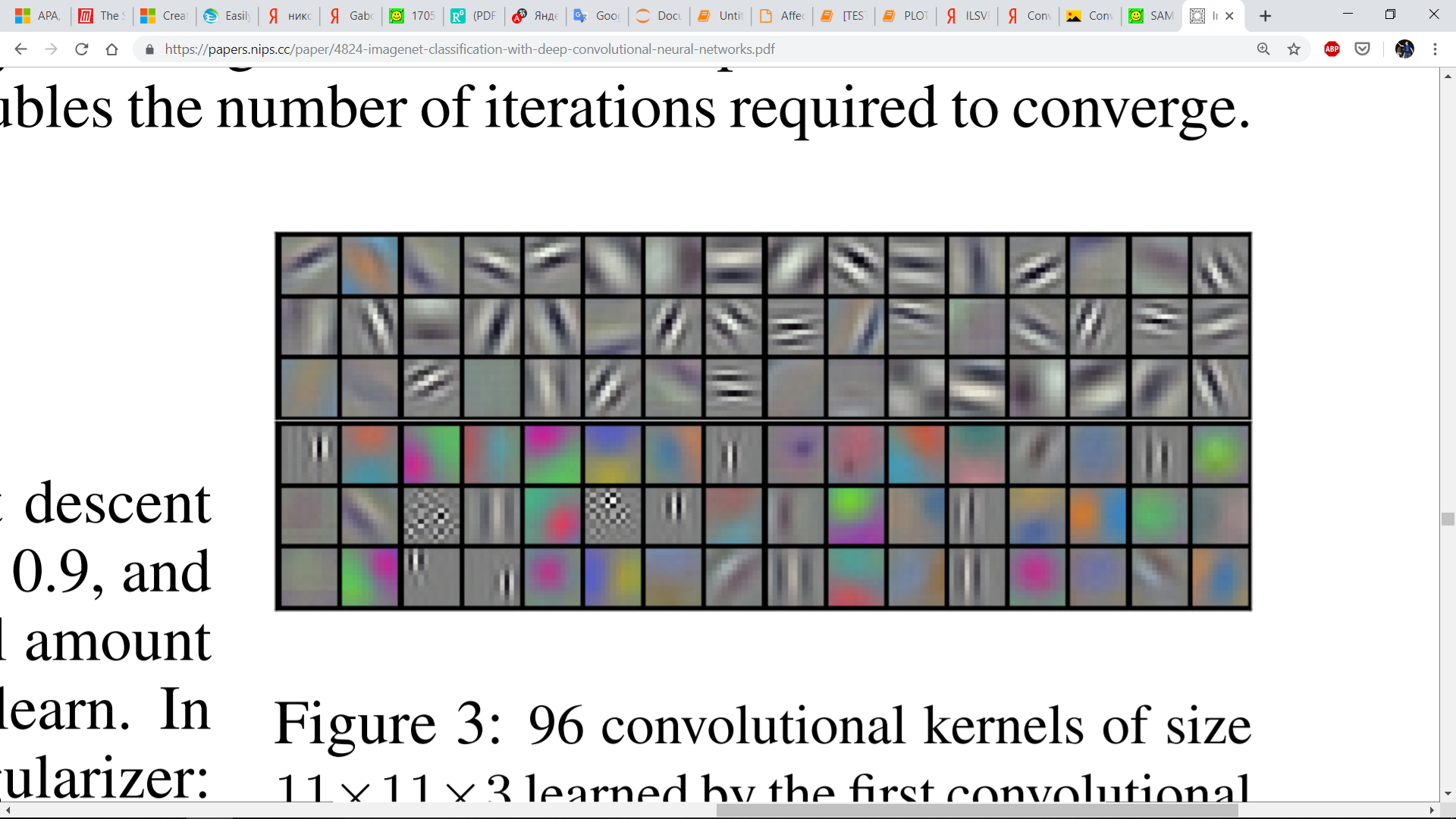


Fig 2 96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images in Alexnet

# Experiments

In this section, the details about datasets and CNNs architectures used during the experiments are explained. Every CNN architecture was implemented in two ways: regular CNN and CNN with Gabor Layer as the first layer of the network (Gabor CNN or GCNN). Serviceability of GCCN has firstly tested on Dogs vs Cats dataset [13]. The further evaluation of GCCN was carried out using AffectNet [14] and IMAGENET ILSVRC2012[15]**.**

## Dogs vs Cats

This dataset consists of 25000 colored images of different size. For validation purposes30% of the dataset was used. All experiments were performed with a batch size of 64. Adam [16] was used as an optimization algorithm for both CNN and GCNN architectures. Pooling [17] and dropout [18] layers, as well as ReLU [19] activation function, were also used. Images were normalized. Table 1 shows the architecture of CNN and GCNN.

Table 1. Architecture of CNN and GCNN used for Dogs vs Cats dataset

|  |  |
| --- | --- |
| Layer | Channels (Output) |
| Convolution (CNN)/Gabor Layer (GCNN)  ReLU  MaxPooling | 32 |
| Convolution  ReLU  MaxPooling | 64 |
| Convolution  ReLU  MaxPooling | 128 |
| Convolution  ReLU  MaxPooling | 128 |
| Convolution  ReLU  MaxPooling | 128 |
| Fully Connected  ReLU | 128 |
| Fully Connected  ReLU | 128 |
| Fully Connected | 2 |

Both networks were trained for 100 epochs. Learning rate was set to 0.001 and betas = (0.9, 0.999). Results are shown in Figure 3. As it can be seen GCNN outperforms regular CNN and converged on earlier epochs. The performance gap achieves 6% accuracy in the end. The performance of GCNN and CNN is listed in detail in table 2.

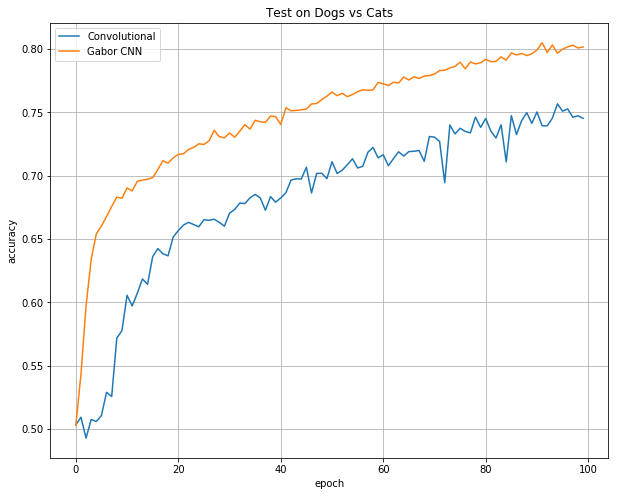
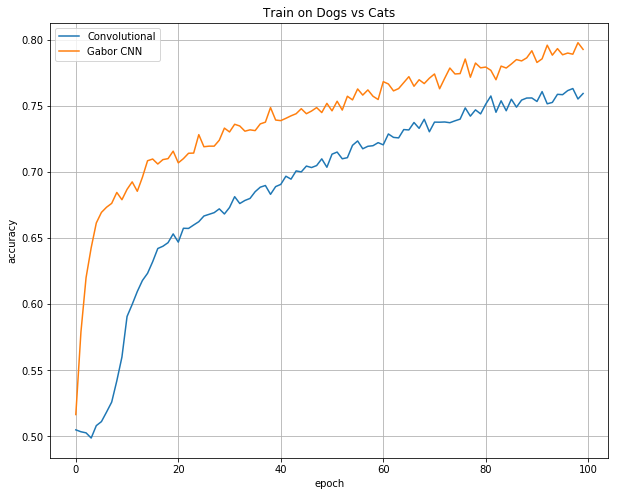


Fig 3 The performance of GCNN and CNN on Dogs vs Cats dataset

Table 2 Accuracy score on Dogs vs Cats dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Train | | Test | |
| Epoch | CNN | Gabor CNN | CNN | Gabor CNN |
| 1 | 0.506 | 0.503 | 0.503 | 0.517 |
| 3 | 0.520 | 0.597 | 0.515 | 0.620 |
| 10 | 0.613 | 0.682 | 0.616 | 0.679 |
| 40 | 0.674 | 0.747 | 0.668 | 0.739 |
| 90 | 0.732 | 0.796 | 0.726 | 0.792 |

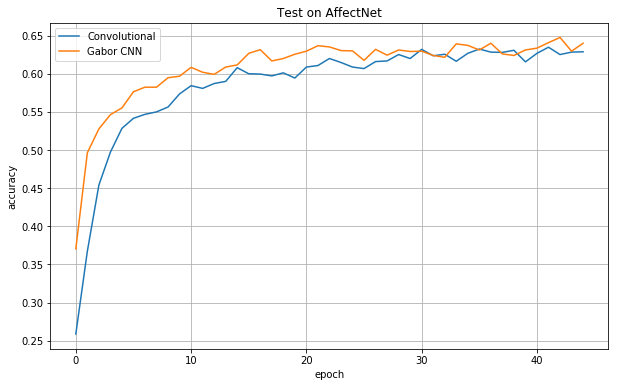
## AffectNet

The AffectNet dataset consists of 420 299 colored images of human emotions of different size. For evaluation only 5 classes were used (Neutral, Happy, Sad, Surprise, Anger), also they were resampled to be a total of 250 000 images and to the number of images in each class was equal. The structures of CNN and Gabor CNN is shown in Table 3.

Table 3 Architecture of CNN and GCNN used for AffectNet dataset

|  |  |
| --- | --- |
| Layer | Channels (Output) |
| Convolution (CNN)/Gabor Layer (GCNN)  ReLU  MaxPooling | 96 |
| Convolution  ReLU  MaxPooling | 256 |
| Convolution  ReLU  MaxPooling | 384 |
| Convolution  ReLU  MaxPooling | 384 |
| Convolution  ReLU  MaxPooling | 256 |
| Fully Connected  ReLU | 128 |
| Fully Connected  ReLU | 128 |
| Fully Connected | 5 |

Gabor CNN achieves better results on earlier epochs both on a train and on a test. The average difference is 3% of the accuracy score. In addition, Gabor CNN converges several epochs earlier than regular CNN. However, unlike Dogs vs Cats dataset, on later epochs, CNN achieves almost the same accuracy score as Gabor CNN and the average accuracy[[1]](#footnote-1) difference drops to 1%. The performance gap is shown in Figure 4. The performance of GCNN and CNN is listed in detail in Table 4.



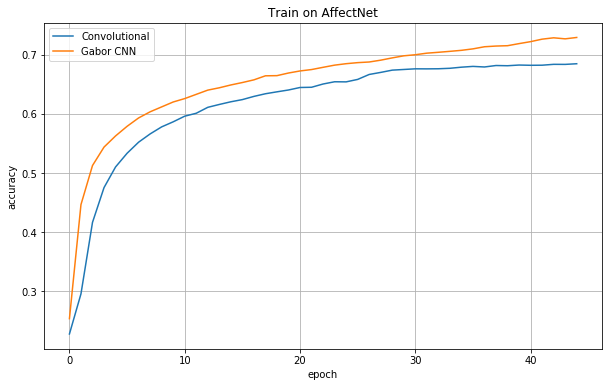


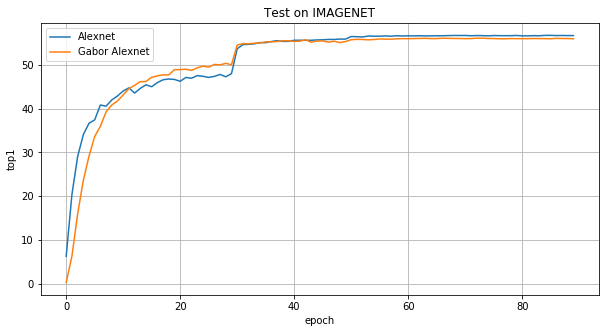
Fig 4 The performance of CNN and Gabor CNN

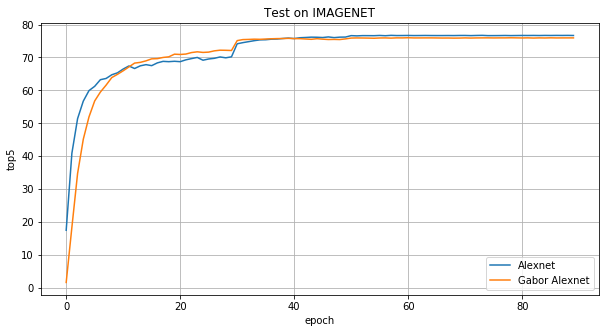
Table 4 Accuracy score on AffectNet dataset

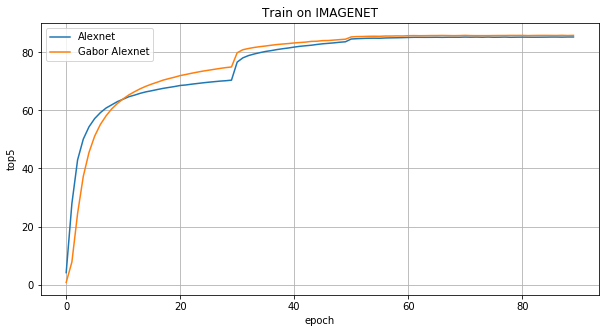
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Train | | Test | |
| Epoch | CNN | Gabor CNN | CNN | Gabor CNN |
| 1 | 0.228 | 0.254 | 0.256 | 0.370 |
| 5 | 0.510 | 0.563 | 0.528 | 0.555 |
| 10 | 0.586 | 0.620 | 0.574 | 0.597 |
| 20 | 0.640 | 0.669 | 0.594 | 0.626 |
| 35 | 0.679 | 0.710 | 0.627 | 0.637 |
| 45 | 0.685 | 0.719 | 0.629 | 0.640 |

## IMAGENET

For evaluation on IMAGENET dataset unchanged dataset from ILSVRC 2012 was used. Data augmentation techniques, such as random horizontal flip and random crop, proposed in [11] was used. Exact version of Alexnet [11] and a special version of Alexnet, which had the first layer changed to Gabor Layer (Gabor Alexnet), were used for the experiment. The learning rate was decreased by factor 10 on 30th and 50th epochs. Unlike on Dogs vs Cats and AffectNet datasets, the network with Gabor Layer had smaller accuracy score on earlier epochs on IMAGENET. However, from 10 to 30epochs the performance gap between Gabor Alexnet and Alexnet achieved almost 2% on top1 and top5 scores in favor of Gabor Alexnet. After reducing the learning rate on 30th epoch Alexnet leveled top1 and top5 scores with Gabor Alexnet. In the end, Gabor Alexnet and Alexnet achieved almost exact results with Gabor Alexnet having fewer parameters due to Gabor Layer structure. The performance of Gabor Alexnet and Alexnet is shown in Figure 5. The performance of Gabor Alexnet and Alexnet is listed in detail in Table 5.







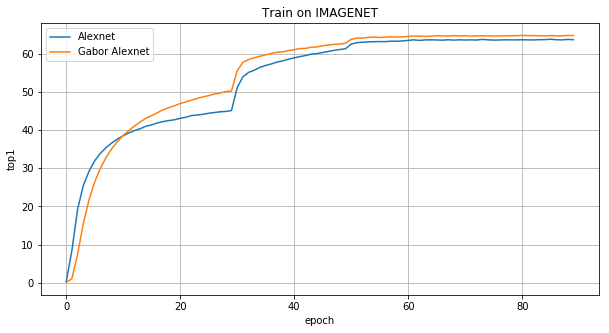


Fig 5 The performance of Alexnet and Gabor Alexnet

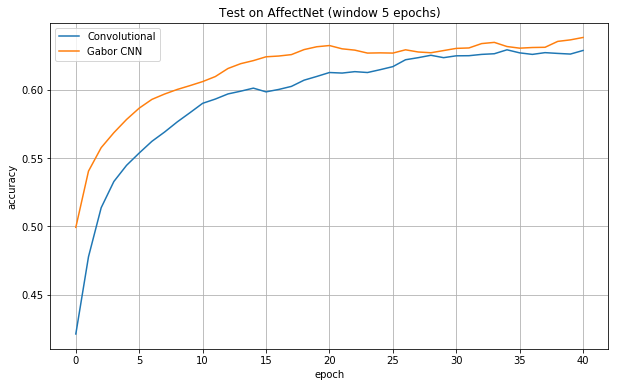
Table 5 The accuracy score in percent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Top1 | | | | Top5 | | | |
| Train | | Test | | Train | | Test | |
| Epoch | Alexnet | Gabor Alexnet | Alexnet | Gabor Alexnet | Alexnet | Gabor Alexnet | Alexnet | Gabor Alexnet |
| 1 | 0.32 | 0.11 | 6.28 | 0.29 | 4.13 | 0.75 | 17.52 | 1.67 |
| 5 | 29.18 | 21.71 | 36.65 | 29.15 | 54.33 | 45.50 | 59.93 | 51.95 |
| 10 | 37.64 | 37.00 | 42.90 | 41.75 | 63.04 | 62.40 | 65.33 | 64.87 |
| 15 | 41.00 | 43.10 | 45.45 | 46.21 | 66.37 | 68.32 | 67.81 | 68.97 |
| 25 | 44.10 | 48.66 | 47.40 | **49.71** | 69.49 | 73.53 | 69.16 | **71.53** |
| 45 | 59.98 | 61.71 | 55.70 | 55.45 | 82.70 | 83.81 | 76.14 | 75.70 |
| 70 | 63.60 | 64.62 | 56.71 | 56.10 | 85.19 | 85.76 | 76.68 | 75.90 |
| 90 | 63.63 | 64.73 | 56.68 | 56.01 | 85.27 | 85.83 | 76.70 | 76.00 |

# Conclusion

This paper has presented a new convolutional layer by incorporating the backpropagation algorithm to Gabor filters, aiming to enhance feature representations and reduce complexity training of DCNNs. The proposed Gabor Layers decrease number of learnable parameters of DCNN. The end-to-end structure of Gabor Layers simplifies its use in popular DCNN architectures. The future work will focus on <…think about it…>.

# Appendix



# References

|  |  |
| --- | --- |
| [1] | K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers:," 6 Feb 2015. [Online]. Available: https://arxiv.org/pdf/1502.01852.pdf. |
| [2] | X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010. |
| [3] | D. Gabor, "Theory of Communication," *Journal of the Institution of Electrical Engineers,* pp. 429-441, 1946. |
| [4] | A. K. Jain, N. K. Ratha and S. Lakshmanan, "Object detection using gabor filters," *Pattern Recognition,* vol. 30, no. 2, pp. 295-309, 1997. |
| [5] | Huang, L.-L. &. Shimizu, A. &. Kobatake and Hidefumi, "Classification-Based Face Detection Using Gabor Filter Features," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2014. |
| [6] | K. Bogdan, "Face Detection Using Convolutional Neural Networks and Gabor Filters," 0302-9743, 2005. |
| [7] | A. Calderón, S. R. Ovalle and J. Victorino, "Handwritten Digit Recognition using Convolutional Neural Networks and Gabor filters," 2003. |
| [8] | S. Luan, B. Zhang and S. Zhou, "Gabor Convolutional Networks," 29 Jan 2018. [Online]. Available: https://arxiv.org/pdf/1705.01450.pdf. |
| [9] | H.-B. Deng, L.-W. Jin, L.-X. Zhen and J.-C. Huang, "A New Facial Expression Recognition Method Based on Local Gabor Filter Bank and PCA plus LDA," *International Journal of Information Technology,* vol. 11, no. 11, pp. 86-96, 2005. |
| [10] | D. Rumelhart, G. Hinton and R. J. Williams, "Learning internal representations by error propagation," in *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*, MIT Press Cambridge, 1986, pp. 318-362. |
| [11] | A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS2012*, 2012. |
| [12] | S. Meshgini, A. Aghagolzadeh and H. Seyedarabi, "Face recognition using Gabor filter bank, kernel principal component analysis and support vector machine," *International Journal of Computer Theory and Engineering,* vol. 4, pp. 767-771, 2012. |
| [13] | Kaggle Inc, "Dogs vs. Cats," 2014. [Online]. Available: https://www.kaggle.com/c/dogs-vs-cats. |
| [14] | M. Mahoor, "AffectNet," 2017. [Online]. Available: http://mohammadmahoor.com/affectnet/. |
| [15] | Stanford Vision Lab, Stanford University, Princeton University , "ImageNet," 2012. [Online]. Available: http://www.image-net.org/. |
| [16] | D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference for Learning Representations*, San Diego, 2015. |
| [17] | J. Nagi, F. Ducatelle, G. A. D. Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber and L. M. Gambardella, "Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition'," in *IEEE International Conference on Signal and Image Processing Applications*, 2011. |
| [18] | G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012. |
| [19] | X. Glorot, A. Bordes and Y. Bengio, "Deep Sparse Rectifier Neural Networks," 2011. |

1. Due to oscillating values of accuracy score, moving average with window size of 5 epochs was used. The plots of moving average for testset is in the appendix. [↑](#footnote-ref-1)