GABOR BINARY LAYER IN CONVOLUTIONAL NEURAL NETWORKS

Chenzhi Jiang, Jianbo Su

Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

Convolutional neural networks(CNNs) have achieved overwhelming success in image recognition. For all the architectures of CNNs, the low-level features extracted from the input is essential for the whole model because it determines the upper bound of accuracy. However, due to the random initialization and end-to-end training mechanism, there is no guarantee that the learned convolutional layer is an excellent low-level feature extractor. Thus, we propose **Gabor binary layer(GBL)** as a more efficient alternative to the first convolutional layer in CNNs. The design principle of GBL is motivated by local Gabor binary patterns. The GBL is comprised of a module of pre-defined Gabor convolutional filters in different orientations and shapes, and a module of fixed randomly generated binary convolutional filters to encode the Gabor features. By replacing the first layer of Resnet-56, Resnet-110, and LBCNN with the GBL, we achieve better performances on SVHN, CIFAR-10 and CIFAR-100 than that of the original models. Results show that the proposed GBL outperforms the standard convolutional layer for extracting low-level features, and thus the GBL can easily improve the performances of CNNs on image recognition.

***Index Terms*—** Gabor, low-level features, convolutional network, image recognition.

1. INTRODUCTION

Deep convolutional neural networks [1, 2, 3, 4] usually integrate low/mid/high-level features [5] and classifier into a holistic object. Convolutional networks are aimed to deal with large amount of low-level features [6]. The low-level features are transformed into distinctive mid/high-level features layer by layer. The transformed features are used for various computer vision tasks. Therefore, the low-level features are so essential that they can influence the performance of entire model.

In CNNs, the first convolutional layer usually plays the role of low-level feature extractor. However, some deep visualization studies [5, 7] show that, the filters in the first layer of Alexnet [1] are repetitive and deficient. These filters are trained by the gradients with backpropagation method[8]. Whereas the gradients have been interacted by a huge amount of uncertain parameters before reaching the first layer. And the gradients are calculated based on high-level features, which actually is varying transformation of the low-level features provided by first layer. Hence, the low-level features are significant and it is uncertain that the first layer of CNNs can provide complete and efficient low-level features.

In traditional computer visual domain, there are splendid algorithms focusing on constructing an excellent low-level feature extractor, such as Gabor [9], local binary patterns(LBP) [10], scale invariant feature transform(SIFT) [11]. Among them, the Gabor feature biologically originates from the responses of the early visual system in mammalian [12, 13, 14]. It has been widely applied to image processing and derives lots of mature and valuable Gabor-based algorithms [15, 16, 17, 18]. The local Gabor binary patterns(LGBP) [15] and local Gabor XOR patterns(LGXP) [16] are milestones and present really high accuracy in face recognition task. Therefore, we infer that the Gabor-based filters can produce an excellent low-level features description of the natural images.

Comparing with the CNNs, these traditional methods are deficient in transforming low-level features into distinctive mid/high-level features. However, traditional methods probably provide some better low-level features than CNNs. Hence, our idea is to transfer and generalize some Gabor-based algorithms to the convolutional neural networks, constructing a more intelligent convolutional layer to extract low-level features from natural images.

In this paper, we design a Gabor binary layer(GBL) that inspired by LGBP and LGXP. The GBL constitutes a Gabor convolutional module(GCM) and a binary convolutional module(BCM). The GCM is a set of pre-defined Gabor filters with different orientations and shapes. The BCM is a set of fixed sparse binary filters to encode the raw Gabor features produced by GCM. The GBL can be applied to any CNNs just by replacing the first layer of CNNs with GBL. So the GBL brings tiny change on structure and computation to the original CNNs. But it can easily improve the performance of CNNs on image recognition. We apply our GBL to some CNNs and present comprehensive experiments on SVHN [19], CIFAR-10 and CIFAR-100 [20]. The experimental results demonstrate the GBL is an excellent low-level feature extractor. It also indicates that the current CNN learning method, which optimizes a holistic loss function by backpropagation, have not produce an outstanding low-level feature extractor.

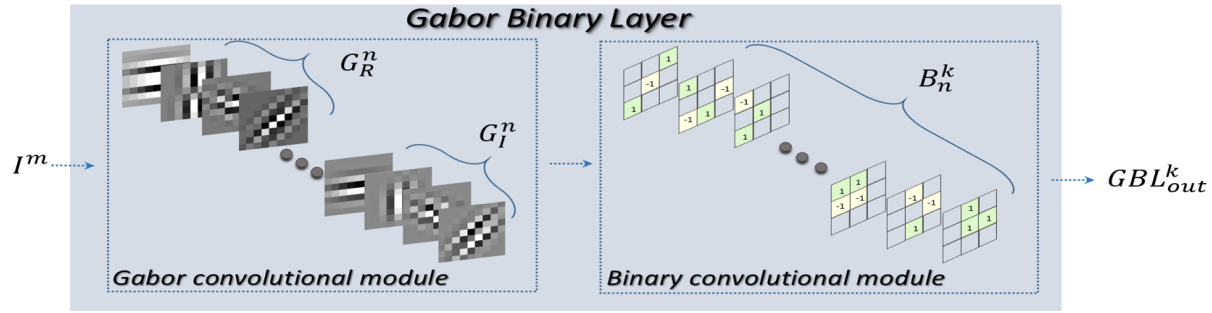

Fig. 1. The architecture of Gabor Binary Layer, including a Gabor convolutional module(GCM) and a binary convolutional module(BCM).

2. GABOR BINARY LAYER

The idea of Gabor binary layer is to implement binary coding on the raw Gabor feature maps. So the GBL consists of a Gabor convolutional module(GCM) and a binary convolutional module(BCM) as shown in Fig 1.

2.1. Gabor convolutional module(GCM)

The Gabor feature maps of a 2D image are defined as the convolutions of an image with Gabor filters, i.e:

$$G(z) = I(z) * \Psi(z), \tag{1}$$

where $I(\bullet)$ denotes the input image, z denotes the pixel, i.e $z = (x, y)$, $*$ denotes the convolution operation and Ψ denotes the Gabor filters defined as follows:

$$\Psi(x, y) = e^{(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})} e^{i(2\pi\frac{x'}{\lambda}+\varphi)}, \tag{2}$$

where

$$x' = x\cos\theta + y\sin\theta, \tag{3}$$

$$y' = -x\sin\theta + y\cos\theta. \tag{4}$$

λ represents the wavelength of the sinusoidal factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, φ is the phase offset, σ is the standard deviation of the Gaussian envelope and γ is the spatial aspect ratio. Changing the value of θ, λ, γ can change the orientation, shape and many details of Gabor filters, which contributes to extract efficient and irredundant features from natural images.

It is obvious that $\Psi(x, y)$ is a complex number, which contains a real component ψ_r and imaginary component ψ_i as follows:

$$\psi_r = e^{(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})} \cos(2\pi\frac{x'}{\lambda} + \varphi). \tag{5}$$

$$\psi_i = e^{(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})} \sin(2\pi\frac{x'}{\lambda} + \varphi). \tag{6}$$

By changing the value of θ, λ and γ in Eq 5 and Eq 6,we can obtain n pairs of Gabor real component filters $G_R^n = [\psi_r^1, \psi_r^2, \cdots, \psi_r^n]$ and Gabor imaginary component filters $G_I^n = [\psi_i^1, \psi_i^2, \cdots, \psi_i^n]$ with different orientations and shapes, which makes up a GCM of $2n$ channels:

$$\begin{aligned} GCM^{2n} &= [G_R^n, G_I^n] \\ &= [\psi_r^1, \psi_r^2, \cdots, \psi_r^n, \psi_i^1, \psi_i^2, \cdots, \psi_i^n]. \end{aligned} \tag{7}$$

The value of θ, λ and γ are equal for each pair of ψ_r^j and $\psi_i^j, j \in [n]$, varying with j. For a multi-channels input image I^m, the output of GCM can be represented as:

$$GCM_{out}^{2n} = \sum_m GCM_m^{2n} * I^m. \tag{8}$$

2.2. Binary convolutional module(BCM)

Here, we use a part of the local binary convolution(LBC) that proposed in LBCNN [21] as the binary convolutional module(BCM). The LBC is motivated by local binary patterns(LBP). So the BCM is a set of fixed pre-defined sparse binary convolutional filters$B_n^k, i \in [k]$. There are only $1, -1$ and 0 in BCM. The percentages of number 1 and -1 in BCM are denoted as sparsity s(i.e 0.5 or 0.9). The BCM is randomly initialized according to the given sparsity. For a multi-channels input x^n, the response of BCM can be expressed as:

$$BCM_{out}^k = \sum_k B_n^k * x^n. \tag{9}$$

2.3. Forming Gabor Binary Layer(GBL)

In order to implement local binary coding on raw Gabor features, we directly connect BCM to GCM to construct

our Gabor Binary Layer(GBL). the response of GBL to a m-channels input image I^m can be expressed as:

$$\begin{aligned} GBL^k_{out} &= BCM^k_{out}(GCM^{2n}_{out}) \\ &= \sum_s B^k_{2n} * (\sum_m GCM^{2n}_m * I^m). \end{aligned} \tag{10}$$

The output of GBL can be seen as many differential maps of raw Gabor features.

2.4. Training with Gabor Binary Layer

To apply GBL to any existing CNN, we only need to replace the first layer of CNN with GBL. The obtained new networks are named as GBL-constructed networks. There are two ways to train with the GBL-constructed networks: 1) Fully-fixed mode. Both GCM and BCM are fixed all the time. There is no parameter updating in the GBL. 2) Semi-fixed mode. The BCM is fixed all the time while the GCM is trainable in the training time. We present the experimental results of these two training modes in the following section.

3. EXPERIMENTAL RESULTS

Theoretically, the GBL can be easily applied to any existing CNNs architectures by replacing the first layer of CNNs with GBL. Here, we apply GBL to two typical deep networks to obtain the GBL-constructed networks. One is the local binary convolutional neural networks(LBCNN) [21] proposed recently, which achieves competitive performance with significant computational savings. Another is a mature and widely acknowledged network, called Resnet [4]. It behaves outstandingly on many datasets for image detection, segmentation and recognition tasks. We evaluate the GBL-constructed networks on three different visual datasets, SVHN, CIFAR-10, and CIFAR-100.

Implementation Details: To ensure a fair comparison between the GBL-constructed networks and the original networks, we only replace the first layer in CNNs with GBL and keep all the rest layers unchanged. Meanwhile, the input and output channels of GBL are equal to that of CNNs. We also use exactly the same data, learning rate decay schedule, optimization method, and iterations as the original networks. There are two modes for training GBL. One is the semi-fixed mode, which means that the BCM is fixed and the GCM is updating during the training period. The other is the fully-fixed mode, which means that both the BCM and the GCM are fixed during the training period.

3.1. SVHN

SVHN [19] is a dataset for classifying real-world house numbers. It contains a training set of 73k and a testing set of 26k 32×32 color images ranging from digit '0' to '9'. Here, we apply GBL to LBCNN to get the new network GBL-LBCNN.

Method	Accuracy(%)
LBCNN [4]	94.50
GBL-LBCNN(fully-fixed)	**95.34**
GBL-LBCNN(semi-fixed)	95.15

Table 1. Classification accuracy on SVHN.

Notice that LBCNN has significant computational savings, so our focus is on improving the performance based on LBCNN structure, but not on pushing the state-of-the-art results.

We construct GBL as follows. The GCM is automatically generated with different orientations(θ) and ellipticity(γ). The wavelength(λ) is set as 3 according to the size of the input image. The Gabor filter size is set as 9×9 with a stride of 2 based on the results shown in Fig 2. The BCM is generated following the rules in LBCNN [21]. Binary filters are randomly initialized with a sparsity of 0.9 and will not be updated during the whole training procedure. The results are shown in Table 1.

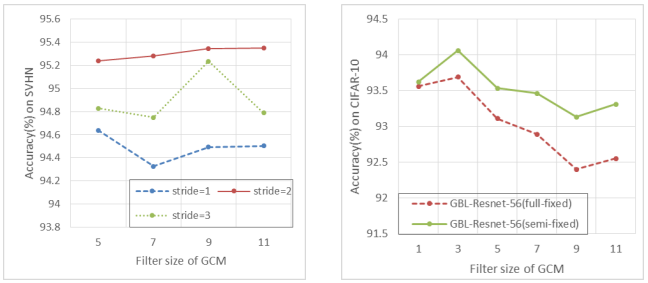

Fig. 2. (L) The accuracy of GBL-LBCNN with varying Gabor filter sizes and strides(fully-fixed mode). (R) The accuracy of GBL-Resnet with varying Gabor filter sizes. Stride is 1.

3.2. CIFAR-10

CIFAR-10 dataset [20] is composed of a training set of 50k and a testing set of 10k 32×32 color images including10 classes. Here, we implement GBL on Resnet to build the GBL-Resnet. The depth of 56 and 110 are tested, so correspondingly there are GBL-Resnet-56 and GBL-Resnet-110. These GBL-constructed models are trained with fully-fixed and semi-fixed training mode. We use the same data and criterions as Resnet. The only difference is that our models start with a lower learning rate. If we use the same initial learning rate as Resnet, which is 0.1, our models will soon diverge. So we use the initial learning rate of 0.04 for fully-fixed training mode, 0.07 for semi-fixed training mode.

Experiment are conducted with $n = \{2, 4, 8, 16\}$, leading to channels of $4, 8, 16, 32$ in GCM. The GBLs are applied to Resnet-56 and trained with fully-fixed training mode. Fig 3 shows the behaviors of the GBL-Resnet-56. The GBL benefits from the increasing channels in GCM till 16. Therefore,

the channel of GCM is set as 16 in the following experiments. We also compare the performances of GBL-Resnet-56 with different Gabor filter sizes in GCM as shown in the right of Fig 2. The experimental results show that filter size of 3×3 is the best choice. Although increasing filter size can include more Gabor wavelets, the pictures in CIFAR-10 are too tiny and of high frequency. So a smaller filter size is more suitable.

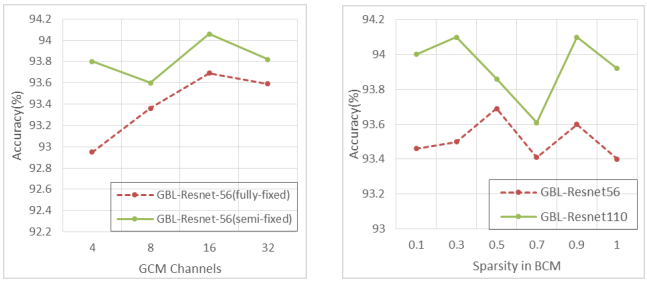

Fig. 3. Accuracy on CIFAR-10. (L) GBL-Resnet-56 with varying channels of the GCM. (R) GBL-Resnet-56 and GBL-Resnet-110 with sparsity changing from 0.1 to 1.

We further explore different sparsities in BCM. The sparsity denotes the percentage of non-zero parameters in the total parameters of BCM. It can affect the complexity of binary coding on raw Gabor features and thus affect the final performances. The results are shown in Fig 3. The accuracy is not linear with sparsity. Sparsity 0.5 is the best choice for Resnet-56 but not for Resnet-110. However, sparsity 0.9 is a nice choice for both the Resnet-56 and Resnet-110. So in the following experiments, the sparsity in BCM is set as 0.9 because it is more stable and performs well on Resnet-56 and Resnet-110.

Finally, a simple and effective architecture of GBL is designed as follows. The GCM is $16 \times 3 \times (3 \times 3)$ and the BCM is $16 \times 16 \times (3 \times 3)$ with sparsity of 0.9. We replace the first layer of Resnet-56 and Resnet-110 with our GBL and train the GBL-Resnet on CIFAR-10 with both fully-fixed and semi-fixed mode. Table 2 compares the performances of our models, Resnet, and some other deep networks. Our GBL-constructed models can achieve significant improvements on Resnet-56 and Resnet-110. The GBL-Resnet-56 with fully-fixed training mode improves the accuracy by 0.68%, which demonstrates that the fixed predefined GBL already can extract excellent low-level features. The GBL-Resnet-56 with semi-fixed training mode improves the accuracy by 0.99%. It shows that the GBL can perform better by updating the parameters in Gabor filters with SGD algorithm.

3.3. CIFAR-100

The CIFAR-100 is similar to the CIFAR-10 except that it contains 100 classes with 600 images for each. And for each class, there are 500 training images and 100 testing images. Here we use exactly the same architecture of GBL as we use on CIFAR-10. The GCM is $16 \times 3 \times (3 \times 3)$ and the BCM is $16 \times 16 \times (3 \times 3)$ with sparsity of 0.9. We also implement GBL on Resnet-56 and Resnet-110 and train the GBL-Resnet with fully-fixed and semi-fixed training mode. Table 3 compares the results.

Method	Accuracy(%)
Resnet-56 [4]	93.03
GBL-Resnet-56(semi-fixed)	**94.02**
GBL-Resnet-56(fully-fixed)	93.63
Resnet-110 [4]	93.57
GBL-Resnet-110(semi-fixed)	**94.25**
GBL-Resnet-110(fully-fixed)	94.05
LBCNN [21]	93.66
MaxMin [22]	92.40

Table 2. Classification accuracy on CIFAR-10.

Method	Accuracy(%)
Resnet-56	72.64
GBL-Resnet-56(semi-fixed)	**72.90**
GBL-Resnet-56(fully-fixed)	71.34
Resnet-110	73.34
GBL-Resnet-110(semi-fixed)	**74.13**
GBL-Resnet-110(fully-fixed)	73.18

Table 3. Classification accuracy on CIFAR-100. The results of Resnet are obtained with our implementation of the source code of Resnet.

4. CONCLUSION

Motivated by traditional local Gabor binary patterns, in this paper, we proposed Gabor binary layer(GBL) as an alternative low-level feature extractor in standard convolutional neural networks. Based on any existing CNNs, one can easily build GBL-constructed networks by replacing the first convolutional layer in CNNs with GBL. So the GBL brings little change in structure and computation to the original CNNs. However, it can significantly improve the performance of CNNs. We apply GBL to LBCNN, Resnet-56, and Resnet-110. All these experimental results on SVHN, CIFAR-10, and CIFAR-100 show that the GBL-constructed networks outperform the original CNNs. It proves that, in terms of extracting the low-level features from natural images, the proposed Gabor binary layer can beat the standard convolutional layer that learned in the state-of-the-art CNNs. Hence, the GBL is a more efficient layer to extract useful and complete low-level features for deep convolutional neural networks.

5. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., Lake Tahoe, Nevada, USA, 2012.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[3] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015, pp. 1–9, IEEE Computer Society.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[5] Matthew D. Zeiler and Rob Fergus, *Visualizing and Understanding Convolutional Networks*, pp. 818–833, Springer International Publishing, Cham, 2014.

[6] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[7] Jason Yosinski, Jeff Clune, Anh Mai Nguyen, Thomas J. Fuchs, and Hod Lipson, "Understanding neural networks through deep visualization," *CoRR*, vol. abs/1506.06579, 2015.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[9] Chengjun Liu and Harry Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition.," *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 467–476, 2002.

[10] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[11] D. G Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] John G Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision research*, vol. 20, no. 10, pp. 847–856, 1980.

[13] S Marĉelja, "Mathematical description of the responses of simple cortical cells," *JOSA*, vol. 70, no. 11, pp. 1297–1300, 1980.

[14] Judson P Jones and Larry A Palmer, "An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex," *Journal of neurophysiology*, vol. 58, no. 6, pp. 1233–1258, 1987.

[15] Wenchao Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang, "Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Oct 2005, vol. 1, pp. 786–791 Vol. 1.

[16] Baochang Zhang, Shiguang Shan, Xilin Chen, and Wen Gao, "Histogram of gabor phase patterns (HGPP): a novel object representation approach for face recognition.," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 57 – 68, 2007.

[17] S. Xie, S. Shan, X. Chen, and J. Chen, "Fusing local patterns of gabor magnitude and phase for face recognition.," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1349–1361, 2010.

[18] Leonardo A. Cament, Luis E. Castillo, Juan P. Perez, Francisco J. Galdames, and Claudio A. Perez, "Fusion of local normalization and gabor entropy weighted features for face identification," *Pattern Recogn.*, vol. 47, no. 2, pp. 568–577, Feb. 2014.

[19] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, 2011, vol. 2011, p. 5.

[20] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.

[21] F. Juefei-Xu, V. N. Boddeti, and M. Savvides, "Local binary convolutional neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 4284–4293.

[22] Michael Blot, Matthieu Cord, and Nicolas Thome, "Max-min convolutional neural networks for image classification," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3678–3682.