Signal, Image and Video Processing
https://doi.org/10.1007/s11760-020-01684-6

ORIGINAL PAPER

Transformation-invariant Gabor convolutional networks

Lei Zhuang[1,2] · Feipeng Da[1,2,3] · Shaoyan Gai[1,2] · Mengxiang Li[4]

Received: 28 October 2019 / Revised: 17 February 2020 / Accepted: 30 March 2020

Abstract
Although deep convolutional neural networks (DCNNs) have powerful capability of learning complex feature representations, they are limited by poor ability in handling large rotations and scale transformations. In this paper, we propose a novel alternative to conventional convolutional layer named Gabor convolutional layer (GCL) to enhance the robustness to transformations. The GCL is a simple but efficient combination of Gabor prior knowledge and parameters learning. A GCL is composed of three components: Gabor extraction module, weight-sharing convolution module, and transformation pooling module, respectively. DCNNs integrated with GCLs, referred to as transformation-invariant Gabor convolutional networks (TI-GCNs), can be easily built by replacing standard convolutional layers with designed GCLs. Our experimental results on various real-world recognition tasks indicate that encoding traditional hand-crafted Gabor filters with dominant orientation and scale information into DCNNs is of great importance for learning compact feature representations and reinforcing the resistance to scale changes and orientation variations. The source code can be found at https://github.com/GuichenLv.

Keywords Gabor filters · Convolutional neural networks · Rotation · Character recognition

1 Introduction

Deep convolutional neural networks (DCNNs) have led to a range of breakthroughs in various fields such as character recognition, object detection, face recognition, and semantic segmentation. However, the learned features are not robust enough to spatial geometric transformations due to the lack of specific modules designed for transformation. Although max pooling layer [2] endows DCNNs with the capacity to process scale changes and moderate rotations, the problem of large rotation and scale transformation cannot be completely solved without transformation encoding mechanism [15].

Numerous state-of-the-art approaches were developed to encode transformation invariance into DCNNs, which can be roughly divided into two categories: transforming the input feature maps and transforming the filters. In [10], the localization network was introduced in spatial transformer networks to predict transformation parameters. The predicted transformation parameters were then used to produce transformed output. Randomly transforming the feature maps during training was introduced in [22] to improve the transform invariance of CNN models. In [6], complicated deformable convolution and deformable RoI pooling were introduced to enhance the transformation modeling capability of DCNNs. However, model parameters and computational complexity were increased. In [25], CNN filters were replaced with complex circular harmonics to offer orientation information.

However, compared with manually designed filters such as Gabor [9], DCNNs blindly learn features from data without prior information and domain knowledge. Therefore, DCNNs-based data-driven feature extraction methods usually require huge training cost and complex model parameters. Visualization of the first layer filters reveals that the convolutional filters are redundantly learned and are significantly similar to Gabor kernels [18]. As a matter of fact, before DCNNs were applied in computer vision tasks, tra-

✉ Feipeng Da
dafp@seu.edu.cn

[1] The School of Automation, Southeast University, Nanjing, China

[2] The Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing, China

[3] Shenzhen Research Institute, Southeast University, Shenzhen, China

[4] The National Research Center of Overseas Sinology, Beijing Foreign Studies University, Beijing, China

Published online: 17 April 2020

ditional Gabor-based algorithms had been widely used in handwritten character recognition, speech recognition [4], and face recognition [1,3]. Considering the fact that traditional hand-crafted Gabor kernels can extract representative features, it is natural to combine Gabor kernels with DCNNs to achieve high performance and significant computational savings. Several works [5,11,24] have combined Gabor and DCNNs by simply using Gabor features as input to a CNN. Some researchers have replaced the first two layers of CNN with fixed or learnable Gabor kernels to reduce computation cost and achieve better network initialization [4,19]. Nevertheless, none of them uses the spatial locality and orientation selectivity characteristics of Gabor filters to design a module to cope with geometric transformations.

In this paper, in order to improve the performance of DCNNs under rotations and scale changes, we propose a new module named Gabor convolutional layer (GCL). To be more exact, in a GCL, the input feature maps are firstly transformed into several Gabor features. Then, all Gabor features are fed into a set of learnable convolutional kernels and are subsequently merged to generate the output features which are robust to spatial transformations. Compared with a conventional convolutional layer, the GCL can easily capture robust features and learn efficient representation under the guidance of Gabor prior information. Meanwhile, GCLs are easily integrated into any deep architecture. DCNNs based on GCLs, which are named as transformation-invariant Gabor convolutional networks (TI-GCNs), can enhance the robustness of learned models against transformations without increasing model parameters.

In summary, the contributions of this paper are:

- A new network module named GCL is proposed to encode scale and orientation information into DCNNs, which can improve the robustness of DCNNs to spatial transformations, such as translations, scale changes, and rotations. GCLs can be easily deployed to any CNN architecture.
- TI-GCNs can get better performance with less parameters by combining multiple Gabor filters with different scales and orientations.
- Experiments on MNIST [17], MNIST with rotations and rescaling [23,27], SVHN [21], and CIFAR [13] datasets are reproduced, achieving better performance on various benchmark than the baseline networks, which shows the ability of TI-GCNs to improve the classification performance and to reinforce the robustness against transformations.

2 Related works

2.1 Gabor filters

Gabor filters have been widely applied to image processing and have derived lots of valuable Gabor-based algorithms since simple cells in the visual cortex of mammalian brains can be modeled by Gabor functions [7]. The Gabor kernels are defined as follows [16]:

$$\psi_{u,v}(z) = \frac{\|k_{u,v}\|^2}{\sigma^2} e^{-(\|k_{u,v}\|^2\|z\|^2/2\sigma^2)}[e^{ik_{u,v}z} - e^{-\sigma^2/2}] \quad (1)$$

where $z = (x, y)$, u and v denote the orientation and scale of the Gabor kernels, respectively, $u \in \{0, 1, \ldots, U - 1\}$, $v \in \{0, 1, \ldots, V-1\}$, $\|\cdot\|$means the norm operator, $\sigma = 2\pi$, and the wave vector $k_{u,v}$ is defined as follows:

$$k_{u,v} = k_v e^{i\phi_u} \quad (2)$$

where $k_v = k_{\max}/f^v$ and $\phi_u = \pi u/U$. $k_{\max}$ is the maximum frequency, and f denotes the spacing factor between kernels in the frequency domain.

2.2 Transformation invariant feature learning

Deep convolutional neural networks can learn expressive features and have the robustness to moderate transitions, scale changes, and small rotations [27]. The equivariance and equivalence of CNN representations to input image transformations have been studied in [15] and several recent works have tried to encode transformation invariance into deep learning models.

Data augmentation Data augmentation [8] can be used to expand the training dataset by creating modified versions of images. Although data augmentation works well, learning feature representation separately for different versions of the original data requires more network parameters and higher training cost.

TI-POOLING By using parallel network architectures, [14] applies the transform invariant pooling operator before the fully connected layers and learns smaller number of network parameters than data augmentation. This topology only uses the most representative instance for learning and limits the redundancy in learned features. Nevertheless, computational complexity is significantly increased as network goes deeper since forward pass is done multiple times.

Harmonic networks Equivariance to patch-wise translation and 360-rotation were exhibited in Harmonic networks [25] by replacing regular CNN filters with circular harmonics, returning a maximal orientation and response for every receptive field patch. Compared to the baseline CNNs, the model parameters and computational cost are increased.

Gabor convolutional networks In order to enhance the robustness of learned features to orientation changes, the Gabor filters are integrated into learnable weights in GCNs [18] by an element-by-element product operation. However, in comparison with the two-dimensional convolution filters in DCNNs, the learned kernels in GCNs are three-dimensional to encode the orientation channel, which actually increases the number of parameters.

The framework we present in this paper is completely different from existing works. Instead of designing new complicated convolution kernels, we construct a new module based on Gabor characteristics to replace conventional convolutional layer. The detail of our work is described in the following section.

3 Transformation-invariant Gabor convolutional networks

Transformation-invariant Gabor convolutional networks (TI-GCNs) are deep convolutional neural networks that replace convolutional layers in DCNNs with Gabor convolutional layers (GCLs). The GCL is an efficient combination of Gabor prior knowledge and parameters learning. DCNNs integrated with GCLs can easily learn robust feature representations to reinforce the resistance to scale changes and orientation variations.

In what follows, we describe the components of the GCL and how to incorporate GCLs into DCNNs to enhance the robustness to transitions, scale changes and rotations. Firstly, we describe the mechanism and the topology of the GCL. Secondly, we show how convenient it is to update the parameters of a GCL during the back-propagation stage. Thirdly, we illustrate how to build TI-GCNs based on baseline CNN.

3.1 Gabor convolutional layer

In this section, we describe the formulation of the GCL, which is the key module of TI-GCNs to learn robust feature representation. The GCL mechanism is split into three parts, as shown in Fig. 1. First, different Gabor kernels are used to extract Gabor features that display orientation and scale selectivity. Then the Gabor features are fed to parallel weight-sharing convolution module to learn more representative features, followed by a transformation pooling module to obtain robust transformation-invariant features. The GCL can be subdivided into rotation-invariant Gabor convolutional layer (RI-GCL), scale-invariant Gabor convolutional layer (SI-GCL), and transformation-invariant Gabor convolutional layer (TI-GCL). These three components are described in the following in detail.

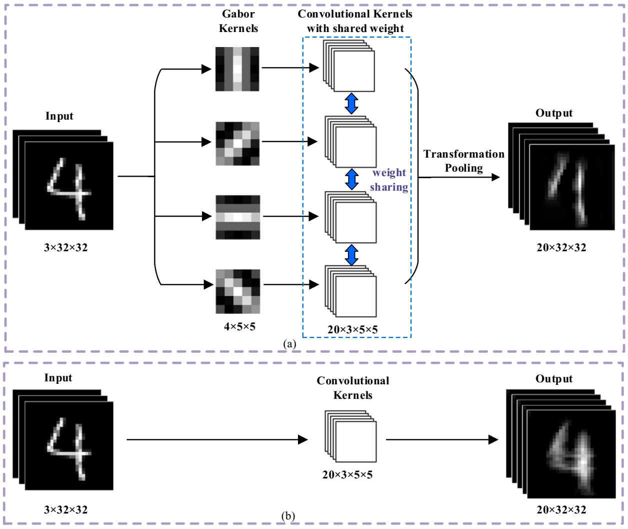

Fig. 1 The comparison between Gabor convolutional layer and conventional convolutional layer. **a** The architecture of a Gabor convolutional layer. The input feature map is passed to Gabor extraction module which consists of Gabor kernels with different orientations and scales. Then parallel weight-sharing convolution module is used to learn more discriminative representations based on the extracted Gabor features. Finally, the transformation pooling module is introduced to obtain robust features against scale changes and rotations. Because of the shared-weights and proposed pooling operation, Gabor convolutional layer has the same parameters and output size as a traditional convolutional layer. **b** the standard convolutional layer

3.1.1 Gabor extraction module (GEM)

In order to utilize spatial locality, scale and orientation selectivity of Gabor kernels, Gabor features, the convolution of the input feature map with a family of Gabor kernels are used as prior information for following parameter learning. Gabor feature is defined as follows [16]:

$$O_{u,v}(z) = I(z) * \psi_{u,v}(z) \tag{3}$$

where $*$ denotes the convolution operator, $I(z)$ means the gray level distribution of an image, and $O_{u,v}(z)$ is the convolution result corresponding to the Gabor kernel at orientation u and scale v. Therefore, The output feature map of the Gabor extraction module (GEM) $\widehat{F}_{\text{gem}}$ can be defined as:

$$\widehat{F}_{\text{gem}}^{(n)} = \widehat{F}_{\text{in}}^{(n)} * G \tag{4}$$

where $\widehat{F}_{\text{in}}^{(n)}$ denotes the nth channel of the input feature map $\widehat{F}_{\text{in}}$, $\widehat{F}_{\text{gem}}^{(n)}$ denotes the nth channel of the output feature $\widehat{F}_{\text{gem}}$, and G refers to a series of Gabor kernels used in the GEM, $G = \{\psi_{u,v}(z)\colon u \in \{0, \ldots, U-1\}, v \in \{0, \ldots, V-1\}\}$. Thus, the number of Gabor kernels is $M = U \times V$. Let the size of the input feature map be $C_{\text{in}} \times W \times W$, where $W \times W$ is the size of the input feature and C_{in} refers to the channel, the size of the output feature map is $C_{\text{in}} \times M \times W \times W$. The output feature of the GEM can be seen as M Gabor features and the size of each Gabor feature is $C_{\text{in}} \times W \times W$.

3.1.2 Weight-sharing convolution module (WCM)

The weight-sharing convolution module (WCM) is composed of several parallel convolutional layers which share weights with each other. To guarantee that each Gabor feature passes through a convolutional layer to obtain corresponding output feature map, the quantity of convolutional layers is chosen to be M. The output feature map of each convolutional layer $\widehat{F}_{\text{wcm}}^{(i)}$ is denoted as:

$$\widehat{F}_{\text{wcm}}^{(i)} = \widehat{F}_{\text{gem}}^{i} * C \tag{5}$$

where $\widehat{F}_{\text{gem}}^{i}$ denotes the ith Gabor feature, $i \in \{1, \ldots, M\}$, and C is the learned weight of the convolutional layers (All convolutional layers share the same weights). Thus, $\widehat{F}_{\text{wcm}}$, the output of the WCM is actually the output features of M convolutional layers.

$$\widehat{F}_{\text{wcm}} = (\widehat{F}_{\text{wcm}}^{(1)}, \ldots, \widehat{F}_{\text{wcm}}^{(M)}) \tag{6}$$

Let the size of shared learned weight be $C_{\text{out}} \times C_{\text{in}} \times k_w \times k_w$, the output size of each convolutional layer $\widehat{F}_{\text{wcm}}^{(i)}$ will be $C_{\text{out}} \times W \times W$, and the size of $\widehat{F}_{\text{wcm}}$ will be $C_{\text{out}} \times M \times W \times W$.

3.1.3 Transformation pooling module (TPM)

Inspired by max pooling [2], the transformation pooling module (TPM) is proposed to be applied on the output features of the WCM to obtain transformation-invariant features. Instead of treating all output features of the WCM independently, we only choose the "canonical" feature representation by using element-wise maximum.

$$\widehat{F}_{\text{tpm}} = \max(\widehat{F}_{\text{wcm}}) \tag{7}$$

where $\widehat{F}_{\text{tpm}}$ denotes the output feature map of the TPM, also means the output of the GCL. Therefore, the size of the output features is $C_{\text{out}} \times W \times W$.

To sum up, the GCL achieves the "canonical" feature representation of the input feature map and enhances the robustness to transformations, as well as maintaining the same output size as a standard convolution layer.

3.1.4 GCL parameters setting

In view of the fact that the demand for the robustness against transformations varies in different computer vision tasks, we propose three kinds of GCLs by using Gabor kernels with different characteristics. To be specific, we construct the RI-GCL by choosing Gabor kernels with the same scale but different orientations in the GEM. Similarly, the SI-GCL can be built by selecting Gabor kernels with the same orientation but different scales. Gabor kernels with different orientations and different scales are chosen to design the TI-GCL.

3.2 Parameters updating

Although the forward calculation of the GCL is divided into three stages, the back-propagation (BP) process is easily implemented. There is no parameter updating in the GEM and the TPM, only the learned filters C in the WCM need to be updated. Since the TPM takes the maximum of M features at each spatial location, the BP process for the TPM is just analogous to how BP is done for spatial max pooling. In conclusion, the gradients of the GCL can be simply computed by a minor modification of traditional BP algorithm of a standard convolutional layer.

3.3 Easy integration

TI-GCNs can be easily built based on any deep learning architecture by only replacing conventional convolution layers with GCLs. Figure 2 illustrates how to construct different TI-GCNs based on the same baseline CNN structure.

4 Experiments

In this section, we present the experimental results on different computer vision datasets based on TI-GCNs. In Sect. 4.1, we conduct experiments on the MNIST handwritten dataset [17], MNIST-rot [27], and MNIST-scale [23]. The results show the superiority of TI-GCLs to improve the classification performance and to enhance the robustness to transformations. Afterwards, experiments based on ResNet network architecture are implemented on the Street View House Numbers (SVHN) dataset [21] to further evaluate the performance of TI-GCNs in Sect. 4.2. To show the generalization ability of our networks, in Sect. 4.3, TI-GCNs are applied on the CIFAR-10 and CIFAR-100 [13] for natural image classification task. We run our experiments based on Pytorch with NVIDIA GeForce GTX 1080Ti.

4.1 MNIST

4.1.1 MNIST and MNIST-rot

The Original MNIST dataset is a very typical dataset to verify the performance of the newly introduced algorithms. Each sample in the MNIST dataset is randomly rotated between [0, 2π] to yield MNIST-rot to test artificially introduced varia-

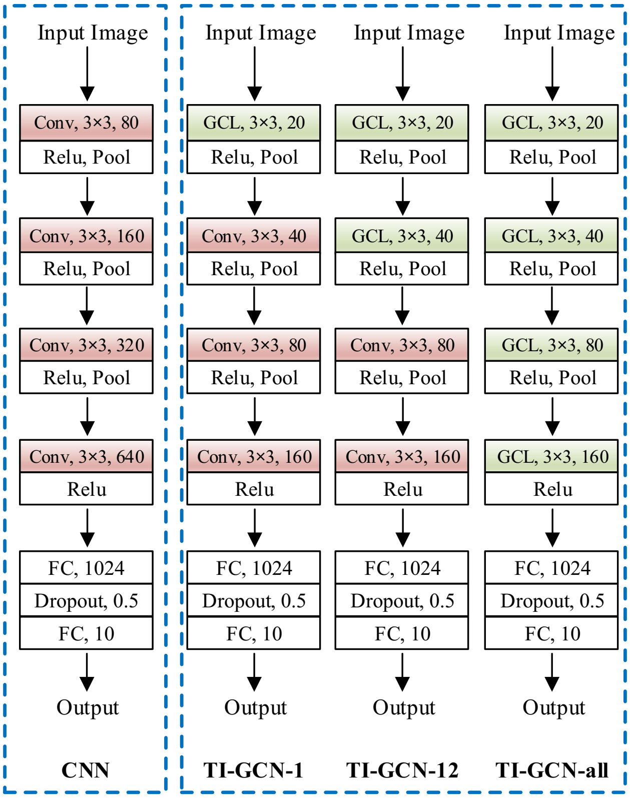

Fig. 2 Network structures of TI-GCNs and baseline CNN. The first layer, the first two layers, and all convolutional layers are, respectively, replaced with GCLs to build TI-GCNs

Table 1 Results on MNIST-rot versus construction methods

Construction method	Accuracy (%)
Baseline CNN	98.34
Replace the first layer	99.30
Replace the first two layers	99.21
Replace all layers	98.71

tions in the data. For each dataset, we randomly selected 50,000 samples from the training set for training and the remaining 10,000 samples for validation. The best model selected by sixfold cross-validation is then applied to the test set. For both datasets we use the same topology described in [27] which consists of four convolutional layers. We perform the training process using SGD with momentum, 128 batch size, and 0.5 dropout rate for fully connected layers.

We first evaluate three construction methods of TI-GCNs on MNIST-rot, where the first layer, the first two layers, all convolutional layers are, respectively, replaced with GCLs. Experimental result in Table 1 implies that extracting Gabor features directly from the image can get the best performance. Placing extra transformation modules at the beginning of the CNN is a more powerful way to learn robust features [22], so we just replace the first convolutional layer in baseline CNNs to construct TI-GCNs in the following experiments.

We choose 4 Gabor kernels with the same scale and different orientations for preliminary test ($M = U = 4, V = 1$). The final results are presented in Table 2. The state-of-the-art STN [10], TI-Pooling [14], ORNs [27], RI-LBCNN [26], and GCNs [18] are used for comparison. Among them, STN builds a spatial transform layer to obtain transformation robust features. TI-Pooling gets the response of main direction by built-in data augmentation and transform-invariant pooling layer. ORNs and RI-LBCNN are built by more expressive filters to encode rotation-invariant features. Gabor filters are incorporated into the convolution filter to improve the robustness of DCNNs to image transformations in GCNs.

In Table 2, the second column shows the width of each convolution layer, and a similar notation is also used in [18,26]. Since the Gabor filters provide spatial and orientation selectivity, we can decrease the width of layer to reduce computational complexity. Training time of an epoch on the original MNIST dataset is listed in the fourth column, which indicates the efficiency of TI-GCNs. The fifth and sixth columns describe the error rates on the original MNIST and the MNIST-rot, respectively. The last column describes the error increase caused by rotations. TI-GCNs achieve better performance with significantly fewer network parameters on the original dataset in comparison with the baseline CNNs. Moreover, on the MNIST-rot datasets, TI-GCNs outperform ORNs, RI-LBCNN, and GCNs, which demonstrates that incorporating Gabor filters into DCNNs indeed helps to enhance the robustness to rotations. It can be observed from the last column of Table 2 that the error rate increase caused by rotations in TI-GCNs is much lower than other state-of-the-art models. It is obvious that we can achieve lower error rate by increasing TI-GCNs model parameters. In conclusion, TI-GCNs enhance feature representations by combining Gabor features which are robust to rotations and thus perform better.

Experiments based on TI-GCNs with 5 × 5 kernels are conducted to further test that how the orientation number of the Gabor kernels influences network performance. The results shown in Table 3 indicate that increasing the number of directions can improve network performance when using 4–8 orientations. However, the classification error will increase when the orientation number exceeds 12. The reason for this phenomenon is that sufficient orientation information is necessary for learning rotation-invariant features, while too much orientation information makes network redundant.

4.1.2 MNIST-scale

MNIST-scale is a variation of the MNIST digit classification benchmark introduced by [23]. We randomly select 10k samples for training, 2k for validation and 50k for testing.

Table 2 Results on MNIST and MNIST-rot

Method	# network stage kernels	# parameters (M)	Time (s)	Original (%)	Rot (%)	Original → Rot(%)
Baseline CNN	80–160–320–640	3.08	3.89	0.73	2.82	286.30
TI-POOLING	80–160–320–640	24.64	31.33	0.97	–	–
STN	80–160–320–640	3.20	4.43	0.66	2.88	336.36
ORN8 (ORAlign)	10–20–40–80	0.96	5.13	0.59	1.42	140.68
ORN8 (ORPooling)	10–20–40–80	0.39	4.22	0.66	1.37	107.58
RI-LBCNN-8	10–20–40–80	0.39	5.40	0.97	1.36	40.21
GCN4 (with 3 × 3)	20–40–80–160	0.78	4.34	0.56	1.28	128.57
GCN4 (with 5 × 5)	20–40–80–160	1.86	15.45	0.48	1.10	129.17
TI-GCN4 (with 3 × 3)	10—20–40–80	**0.13**	**1.66**	0.66	1.05	59.09
TI-GCN4 (with 3 × 3)	20–40–80–160	0.33	2.03	0.61	0.81	32.79
TI-GCN4 (with 3 × 3)	40–80–160–320	0.94	3.57	0.56	0.72	**28.57**
TI-GCN4 (with 5 × 5)	10–20–40–80	0.20	2.97	0.63	0.90	42.86
TI-GCN4 (with 5 × 5)	20–40–80–160	0.60	4.08	0.57	0.77	35.09
TI-GCN4 (with 5 × 5)	40–80–160–320	2.02	7.08	**0.47**	**0.68**	44.68

Bold values indicate the minimum value of experimental results

Table 3 Results on MNIST and MNIST-rot versus orientation number of Gabor kernels

# Gabor orientations	4	8	12	16
Error on MNIST (%)	0.57	**0.49**	0.50	0.55
Error on MNIST-rot (%)	0.77	**0.70**	0.76	0.72

Bold values indicate the minimum value of experimental results

Table 4 Average classification errors and standard deviations over 5 runs on the MNIST-scale

Method	Error (%)	Precision (%)
TIRBM [23]	5.5	–
SI-CNN [12]	3.13 ± 0.19	96.39
3-layer CNN	3.13 ± 0.11	96.28
Scale equivariant CNN [20]	2.44 ± 0.07	96.98
TI-GCN3	**1.81 ± 0.05**	97.74
TI-GCN5	1.83 ± 0.02	97.67

Bold value indicates the minimum value of experimental results

We use an architecture with three convolutional layers, all with 7 × 7 filters, which is the same architecture as [20]. Gabor kernels with 3 or 5 scales are chosen in SI-GCL to build corresponding TI-GCN3 and TI-GCN5, respectively. We report the performance of our algorithm on test set based on the average over 5 runs in Table 4 compared to previously published results on the same dataset.

It is observed from the experiments that TI-GCN3 achieves test error of 1.81% on MNIST-scale and outperforms Scale equivariant 3-layer CNN [20] as well as SI-CNN [12]. The results on MNIST-scale show the fact that pooling over features based on Gabor filters with different scales brings in robust feature representation. TI-GCNs can enhance robustness to scale variations without increasing parameters amount.

4.2 Street view house numbers

The Street View House Numbers (SVHN) dataset[21] is a real-world image dataset obtained from house numbers in Google Street View images. SVHN contains over 600,000 MNIST-like 32x32 digit images centered around a single character. We use 73,257 digits for training, 26,032 digits for testing, to recognize digits and numbers in natural scene images.

Specifically, we replace the first convolution layer with a TI-GCL to build TI-GCN based on ResNet, leading to TI-ResGCN. We build TI-ResGCN-56 and TI-ResGCN-110 to compare the performance with VGG, ResNet, ORNs, and GCNs. The results shown in Table 5 further reflect the superior property of TI-GCNs when used in real-world problems. We can conclude from the results that TI-ResGCNs perform better than baseline ResNets, which proves that the GCL is an efficient alternative to conventional convolutional layer. Table 5 also shows that TI-ResGCNs outperform previous state-of-the art ORN by 0.26% with fewer parameters.

4.3 Natural image classification

In order to further validate whether our framework can handle rotation variations and scale changes in natural image classification task, we evaluate TI-GCNs on the CIFAR datasets [13]. The CIFAR-10 and CIFAR-100 datasets consist of

Table 5 Results on SVHN

Method	VGG	ResNet-56	ResNet-110	GCN	OR-ResNet	TI-ResGCN-56	TI-ResGCN-110
# Parameters (M)	20.3	0.85	1.73	2.2	2.2	0.85	1.73
Accuracy (%)	95.66	95.86	95.80	96.90	96.35	96.61	96.74

Extra dataset is not used for training

Table 6 Classification errors on CIFAR-10 and CIFAR-100

Method	# Paras (M)	CIFAR-10	CIFAR-100
NIN	–	8.81	35.67
VGG	20.1	6.32	28.69
ResNet-56	0.85	6.97	–
ResNet-110	1.73	6.43	25.16
GCN2-110	3.4	5.65	26.14
OR-ResNet	0.9	5.31	–
TI-ResGCN-56	0.85	5.83	25.13
TI-ResGCN-110	1.73	**5.26**	**23.03**

Bold values indicate the minimum value of experimental results

60,000 32 × 32 color images drawn from 10 and 100 classes split into 50,000 training and 10,000 testing images.

TI-ResGCNs for CIFAR-100 are built based on the "bottleneck" building block. Experiments on ResNet, ORNs, and GCNs are used for comparison and analysis. The results listed in Table 6 demonstrate that TI-ResGCN-56 (5.83% 0.85M) with fewer parameters outperforms larger baseline ResNet-110 (6.43% 1.73M) on CIFAR-10, verifying that TI-GCNs can extract more representative features. Splendid performance of TI-ResGCNs on CIFAR-100 can also be observed. The accuracy of TI-ResGCNs on CIFAR-100 reaches to 76.97%, higher than state-of-the-art methods ORNs and GCNs, which further demonstrates the strong capability of our model to deal with real-world spatial transformations.

5 Conclusion

We have proposed an alternative to conventional convolutional layer to enhance model robustness to scale changes and rotation variations. Transformation-invariant Gabor convolutional networks (TI-GCNs) proposed in this paper outperform the baseline DCNNs without increasing model parameters and computational complexity. Experimental results show that TI-GCNs can improve the performance of DCNNs on several real-world classification tasks.

Acknowledgements This work is supported by the Shenzhen Science and Technology Innovation Committee (STIC) under Grant JCYJ20180306174455080.

References

1. Baochang, Z., Shiguang, S., Xilin, C., Wen, G.: Histogram of gabor phase patterns (hgpp): a novel object representation approach for face recognition. IEEE Trans. Image Process. **16**(1), 57–68 (2007)
2. Boureau, Y.L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th International Conference on Machine Learning, pp. 111–118 (2010)
3. Chai, Z., Sun, Z., Mendezvazquez, H., He, R., Tan, T.: Gabor ordinal measures for face recognition. IEEE Trans. Inf. Forensics Secur. **9**(1), 14–26 (2014)
4. Chang, S.Y., Morgan, N.: Robust CNN-based speech recognition with Gabor filter kernels. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 905–909 (2014)
5. Chen, Y., Zhu, L., Ghamisi, P., Jia, X., Li, G., Tang, L.: Hyperspectral images classification with Gabor filtering and convolutional neural network. IEEE Geosci. Remote Sens. Lett. **14**(12), 2355–2359 (2017)
6. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
7. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. J. Opt. Soc. Am. A-Opt. Image Sci. Vis. **2**(7), 1160–1169 (1985)
8. van Dyk, D.A., Meng, X.L.: The art of data augmentation. J. Comput. Graph. Stat. **10**(1), 1–50 (2001)
9. Gabor, D.: Theory of communication. Part 1: the analysis of information. J. Inst. Electr. Eng. III Radio Commun. Eng. **93**(26), 429–441 (1946)
10. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: Advances in Neural Information Processing Systems, pp. 2017–2025 (2015)
11. Jiang, C., Su, J.: Gabor binary layer in convolutional neural networks. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp 3408–3412 (2018)
12. Kanazawa, A., Sharma, A., Jacobs, D.: Locally scale-invariant convolutional neural networks. arXiv preprint arXiv:1412.5104 (2014)
13. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical Report, University of Toronto, Toronto, ON, Canada (2009)
14. Laptev, D., Savinov, N., Buhmann, J.M., Pollefeys, M.: Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
15. Lenc, K., Vedaldi, A.: Understanding image representations by measuring their equivariance and equivalence. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
16. Liu, C., Wechsler, H.: Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Trans. Image Process. **11**(4), 467–476 (2002)
17. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognit. **36**(10), 2271–2285 (2003)

18. Luan, S., Chen, C., Zhang, B., Han, J., Liu, J.: Gabor convolutional networks. IEEE Trans. Image Process. **27**(9), 4357–4366 (2018)
19. Ma, Y., Luo, Y., Yang, Z.: Geometric operator convolutional neural network. arXiv preprint arXiv:1809.01016 (2018)
20. Marcos, D., Kellenberger, B., Lobry, S., Tuia, D.: Scale equivariance in CNNs with vector fields. arXiv preprint arXiv:1807.11783 (2018)
21. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011)
22. Shen, X., Tian, X., He, A., Sun, S., Tao, D.: Transform-invariant convolutional neural networks for image classification and search. In: Proceedings of the 24th ACM International Conference on Multimedia, pp. 1345–1354 (2016)
23. Sohn, K., Lee, H.: Learning invariant representations with local transformations. arXiv preprint arXiv:1206.6418 (2012)
24. Wang, Q., Zheng, Y., Yang, G., Jin, W., Chen, X., Yin, Y.: Multiscale rotation-invariant convolutional neural networks for lung texture classification. IEEE J. Biomed. Health Inform. **22**(1), 184–195 (2018)
25. Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J.: Harmonic networks: dep translation and rotation equivariance. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
26. Zhang, X., Liu, L., Xie, Y., Chen, J., Wu, L., Pietikainen, M.: Rotation invariant local binary convolution neural networks. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (2017)
27. Zhou, Y., Ye, Q., Qiu, Q., Jiao, J.: Oriented response networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.