Geometric Operator Convolutional Neural Network

Yangling Ma  Yixin Luo  Zhouwang Yang
School of Mathematical Sciences, USTC

Abstract

The Convolutional Neural Network (CNN) has been successfully applied in many fields during recent decades; however it lacks the ability to utilize prior domain knowledge when dealing with many realistic problems. We present a framework called Geometric Operator Convolutional Neural Network (GO-CNN) that uses domain knowledge, wherein the kernel of the first convolutional layer is replaced with a kernel generated by a geometric operator function. This framework integrates many conventional geometric operators, which allows it to adapt to a diverse range of problems.

Under certain conditions, we theoretically analyze the convergence and the bound of the generalization errors between GO-CNNs and common CNNs. Although the geometric operator convolution kernels have fewer trainable parameters than common convolution kernels, the experimental results indicate that GO-CNN performs more accurately than common CNN on CIFAR-10/100. Furthermore, GO-CNN reduces dependence on the amount of training examples and enhances adversarial stability. In the practical task of medically diagnosing bone fractures, GO-CNN obtains 3% improvement in terms of the recall.

1 Introduction

Convolutional Neural Networks have been successfully applied in many fields during recent decades, but the theoretical understanding of the deep neural network is still in the preliminary stages. Although Convolutional Neural Networks have strong expressive abilities, they have to two clear deficiencies. First, as complex functional mappings, Convolutional Neural Networks’ generalization ability weakens. This is due to overfitting, which may occur due to the large number of parameters and the large model size. Stemming from these two defects, a great deal of research has been done to modify CNNs (Dai et al. 2017) (Wang et al. 2018) (Sarwar, Panda, and Roy 2017).

Before CNNs were applied, traditional geometric operators had developed quite well. Each geometric operator represents the precipitation of domain knowledge and prior information. For example, the Sobel operator (Works) is a discrete difference operator, which can extract image edge information for edge detection. The Schmid operator (Schmid 2001) is an isotropic circular operator, which extracts texture information from images for face recognition. The Histogram of Oriented Gradients (HOG) (Dalal and Triggs 2005) is a statistic operator of gradient direction, which extracts edge direction distributions from images for pedestrian detection and other uses.

Many computer vision tasks require domain knowledge and prior information. Geometric operators can make use of domain knowledge and prior information, but cannot automatically change parameter values by learning from data. Convolutional Neural Networks have strong data expression abilities and learning abilities, but they struggle to make use of domain knowledge. For better data learning, we have combined the two. It is natural to directly use geometric operators for pre-processing, and then classify the data through a Convolutional Neural Network (Yao et al. 2016). However, this method uses human experience to select geometric operator parameter values, and then carries out the Convolutional Neural Network learning separately. This method is a kind of two-stage technique, and without reducing parameter redundancy in a Convolutional Neural Network, it is difficult to achieve global optimization. The method proposed in this paper directly constructs geometric operator convolution and then integrates geometric operator convolution into a Convolutional Neural Network to form a new framework - the Geometric Operator Convolutional Neural Network. This method achieves global optimizations and utilizes the properties of geometric operators.

In summary, the contributions of this work are as follows:

- This framework can integrates many conventional geometric operators, which reveals its broad customization capabilities when handling diverse problems.
- In theory, the same approximation accuracy and generalization error bounds are achieved when geometric operators meet certain conditions.
- The Geometric Operator Convolutional Neural Network not only reduces the redundancy of the parameters, but also reduces the dependence on the amount of the training samples.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
The Geometric Operator Convolutional Neural Network enhances adversarial stability.

We organize the remaining chapters in the following sequence. We first briefly introduce related work in Sec. 2. In Sec. 3, we describe our framework for the Geometric Operator Convolutional Neural Network, and in Sec. 4, we introduce theoretical analyses. Experiments and conclusion are presented in Sec. 5, Sec. 6, respectively.

2 Related Work

In recent years, Convolutional Neural Networks have been widely used in various classification and recognition applications (Krizhevsky, Sutskever, and Hinton 2012) (Hu et al. 2014). Convolutional Neural Networks have achieved advanced success in various problems. All CNNs adopt an end-to-end approach to learning; however, each unique task is associated with its own distinctive domain knowledge and prior information. Thus, to improve classification accuracy, researchers use priori information that is tailored to each specific task and each specific Convolutional Neural Network. One way to do this is to use the traditional image processing algorithm as a preprocessing step. Another way is to use the traditional image processing algorithm to initialize convolution kernels.

Classification accuracy is a primary concern for researchers in the machine-learning community. Different pre-processing models, such as filters or feature detectors, have been employed to improve the accuracy of CNNs. One example of this is the Gabor filter with CNN (Daugman 1988). The Gabor filter is a feature extractor based on human vision. Besides the Gabor filter, some people also use Fisher vectors (Cimpoi, Maji, and Vedaldi 2014), sparse filter Banks (Pfister and Bresler 2015), and the HOG algorithm (Lu, Wang, and Zhang 2018) combined with a CNN to improve accuracy. Based on the human visual system, these filters are found to be remarkably well-suited for texture representation and discrimination. In the works by Bogdan et al. (Kwolek 2005) and Mounika et al. (Mounika, Reddy, and Reddy 2012), the Gabor filter is used to extract features from the input image in a pre-processing step. However, these methods require a kind of two-stage procedure that may not reach the optimal global solution.

In addition, some scholars use traditional image processing algorithms to initialize convolutional kernels, such as building a Feature Pyramid Network with an image pyramid for multi-scale feature extraction (Lin et al. 2017). Geometric operators are widely used in traditional image processing algorithms. Many researchers use the Gabor filter to fix the first convolution layer, while other layers, which are common convolution layers, can be trained to improve their accuracy (Yao et al. 2016) (Sarwar, Panda, and Roy 2017). Vijay et al. (John, Boyali, and Mita) simultaneously adopted the weight of the first layer convolution with the Gershgorin circle theorem and the Gabor filter constraint to improve the classification accuracy when Convolutional Neural Networks propagated backward. In (Calderón, Roa, and Victorino 2003) (Chang and Morgan 2014), the authors have attempted to get rid of the pre-processing overhead by introducing Gabor filters in the first convolutional layer of a CNN. In addition, some researchers use filters to initialize multiple convolutional kernels. Shangzhen et al. (Lu, Wang, and Zhang 2018) only used the Gabor function to create kernels in four directions to initialize the convolutional kernels from a Convolutional Neural Network. These methods change the initialization weight and use domain knowledge, but they do not reduce the redundancy of model parameters, and they do not enhance the transformation ability of the model.

In summary, the above two ways only use domain knowledge and prior information to improve Convolutional Neural Networks and classification accuracy. In this paper, a new network, the Geometric Operator Convolutional Neural Network, is proposed. This method integrates geometric operators, namely the filters, into a convolutional neural network. This network can not only make use of domain knowledge and prior information, but also reduce the redundancy of network parameters and enhance the ability of model transformation.

3 The framework of the Geometric Operator Convolutional Neural Network

Traditional geometric operators have many properties. By convolving geometric operators and integrating them into a deep neural network to form a Geometric Operator Convolutional Neural Network, we not only retain the characteristics of geometric operators, but also give play to the powerful feature expression ability of deep neural networks. This framework renders image classification tasks more effective. The method’s construction is described in detail in the following section.

3.1 Geometric operators

Before the development of deep Convolutional Neural Networks, traditional image feature extraction methods were based on traditional image processing algorithms, primarily geometric operators. At present, a large number of geometric operators have been applied, such as the Scale Invariant Feature Transform (SIFT) (Lowe 1999), the Roberts operator (Rosenfeld 1981), the Laplace operator (van Vliet, Young, and Beckers 1989), the Gabor operator (Han and Ma 2007), and so on. Each operator has different characteristics. Therefore, different geometric operators are used in different application scenarios, according to the characteristics of each unique problem. For example, SIFT looks for feature points in different scale spaces for pattern recognition and image matching. The Roberts operator uses local differences to find edges for edge detection, and the Laplace operator uses isotropic differentials to retain details for image enhancement.

Geometric operators represent the precipitation of domain knowledge and prior knowledge. The Geometric Operator Convolutional Neural Network is proposed in this paper, which uses the characteristics of geometric operators. The first step in this framework is to convolve geometric operators. In this paper, the Gabor operator and the Schmid operator are mainly used as examples to illustrate how to carry
out convolutions and integrate these convolutions into Convolutional Neural Networks. Other geometric operators in subsequent studies employ similar concepts.

3.2 Convolution of geometric operators

Gabor operator In order to study the frequency characteristics of local range signals, Dennis Gabor (Gabor 1946) proposed the famous “Window” Fourier transform (also called the short-time Fourier transform, STFT) in the paper “Theory of communication” in 1946. This is now known as the Gabor operator; when combined with images, it is referred to as the Gabor filter. Until now, the Gabor filter has undergone many developments, and its primary characteristics are listed below. First, the Gabor filter has the advantages of both spatial and frequency signal processing. As shown in Eqn. 1.0, the Gabor operator is essentially a Fourier transform with a gaussian window. For an image, the window function determines its locality in the spatial domain, so the spatial domain information from different positions can be obtained by moving the center of the window. In addition, since the gaussian function remains the same after the Fourier transform, the Gabor filter can extract local information in the frequency domain. Second, the Gabor filter’s response to biological visual cells may be an optimal feature extraction method. In 1985, Daugman (Daugman 1985) extended the Gabor function to a 2-dimensional form and constructed a 2D Gabor filter on this basis. It was surprising to find that the 2D Gabor filter was also able to obtain both the minimum uncertainty of time and frequency domain at the same time, while maintaining consistency with the mammalian model of retinal nerve cell reception. Third, the Gabor kernels are similar to the convolution kernels from the first convolutional layer in the CNN. An illustration of this similarity is shown in Fig. 1. From the visualization of the first convolutional layer in AlexNet, which was proposed by Alex et al. (Krizhevsky, Sutskever, and Hinton 2012). Some convolution kernels present geometric properties, as in the kernel function from the Gabor filter. From this feature, it can also be explained that there are parameter redundancies in the Convolutional Neural Network, and the Gabor operator can be convoluted and integrated into CNN. Lastly, the Gabor filter can extract directional correlation texture features from an image. As shown in Fig. 2, there are 40 Gabor kernels from five scales and eight directions convolving with an image. Texture feature maps in different directions can be obtained from the original image.

\[ g_{\theta, \phi, \gamma, \sigma, \lambda}(x, y) = \exp \left( -\frac{x'^2 + y'^2}{2\sigma^2} \right) \cos \left( \frac{2\pi x'}{\lambda} + \phi \right) \]

\[ x' = x\cos\theta + y\sin\theta \]
\[ y' = -x\sin\theta + y\cos\theta \]

(1.0)

Since the Gabor operator combines with the CNN in the image, better feature expressions can be obtained. There are two main binding methods. First, the image is preprocessed by the Gabor operator, and then its features are extracted by the CNN. Next, the gabor operator is convoluted to form a convolution layer, and then we integrate this convolution into the common Convolutional Neural Network. The second approach is used in this article. As shown in Eqn. 1.0, the Gabor kernel function has 5 parameters, which are ob-

(a) Convolution kernels from the first layer of ResNet50

(b) Gabor kernels

Figure 1: The similarities between the CNN’s first convolutional kernels and Gabor kernels.

Input  Gabor Kernels  Output

Figure 2: The results of the Gabor operator on an image.
tained by learning and then regenerated into an $m \times m$ kernel. We replace the common convolution kernels with these Gabor kernels to form a convolutional layer. However, for the common convolutional layer, an $m \times m$ convolution kernel is generated by an identity mapping, which requires $m^2$ parameters. So, our method reduces the number of trainable parameters in the convolutional layer.

**Schmid operator** In 2001, Schmid et al. (Schmid 2001) proposed a Gabor-like image filter, namely the Schmid operator. As shown in Eqn. 2.0, its composition is similar to the kernel function of the Gabor operator, so it retains the properties of the Gabor operator. In addition, as shown in Fig. 3, when the original image and a version of that image that has been rotated 90 degrees are both convolved with the same Schmid kernel, the resulting characteristic graph exhibits only 90 degrees of rotation; in other words, the Schmid operator has rotation invariance. The Schmid operator is then convoluted, and we integrate this convolution into common Convolutional Neural Network. This network improves the model’s adversarial stability to rotation and improve the image feature extraction effect. Similar to the convolution of the Gabor operator, as shown in Eqn. 2.0, the Schmid kernel function has two parameters, which are obtained by learning and then generated by the Schmid kernel. Finally, we replace common convolution kernels with Schmid kernels to form a convolutional layer.

$$F_{\sigma,\tau}(r) = \exp\left(\frac{-r^2}{2\sigma^2}\right) \cos\left(\frac{2\pi \tau r}{\sigma}\right)$$  \hspace{1cm} (2.0)

In this paper, only two geometric operator convolutions are explained. Similarly, for other geometric operators, operator kernels are generated by operator kernel functions, which replace common convolution kernels to form a convolutional layer. Due to the diversity of geometric operators, different geometric operators can be replaced with geometric operator convolutions, so the geometric operator convolution is customizable. There is a kind of geometric operator to form any kind of geometric operator convolution. Consequently, a question that must be addressed is how we combine multiple geometric operators with common Convolutional Neural Networks to form the Geometric Operator Convolutional Neural Network.

### 3.3 Geometric Operator Convolutional Neural Network

Only a visualization of the first layer of convolution kernels maintains some geometric characteristics, so the Geometric Operator Convolutional Neural Network proposed in this paper only replaces kernels from the first convolutional layer with geometric operator kernels. The framework of the Geometric Operator Convolutional Neural Network is introduced in Fig. 4. First, kernels from the first convolutional layer are calculated by the parameters of various geometric operators. Then, we concatenate all the calculated convolutional kernels in the last dimension to obtain a complete convolutional kernel. This convolution kernel is used as the weight of the first convolution layer in the Geometric Operator Convolutional Neural Network, and then the common convolution layer and output layer are connected. In this way, we have defined the forward propagation of the whole Geometric Operator Convolutional Neural Network. So, in reverse propagation, the gradient of loss is transferred to the convolution kernel; this process is different from the usual convolution. Here, the convolution kernel generated by the geometric operator needs to further use the chain derivative rule (i.e., Eqn. 3.0, where $L$ is the loss function, $w$ is each convolution kernel, and $p_i$ is the parameter to generate each convolution kernel) to transfer the gradient to the parameters of each convolution kernel. Then, trainable parameters are updated by gradient descent algorithms, and the whole Geometric Operator Convolutional Neural Network is complete.

$$\frac{\partial L}{\partial p_i} = \frac{\partial L}{\partial w} \frac{\partial w}{\partial p_i}$$  \hspace{1cm} (3.0)

### 4 Theoretical analyses

The whole framework of the Geometric Operator Convolutional Neural Network has been introduced above. Next, we
describe how to theoretically analyze the Geometric Operator Convolutional Neural Network. It is theoretically proved that although the number of trainable parameters in the Geometric Operator Convolutional Neural Network decreases, the effectiveness for computer vision tasks does not decrease.

4.1 Definition of data and loss function

- We denote the input by \( S = \{ I_i \}_{i=1}^{N} \), the corresponding label is \( \{ y_i | y_i = 0 \text{ or } 1 \} \). 
- The loss function is Mean Square Error.
- The output of the neural network is \( \hat{y}_i \), for each input \( I_i \), and the empirical loss function is defined as follows:

\[
\hat{E}_S[h] = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 
\]

(4.0)

Lemma 1. (Cybenko 1989) Define a functional class \( \Pi \subset \{ f : f : \mathbb{R}^d \to [0, 1] \} \), where each \( f \in \Pi \) can be approximated with error at most \( \epsilon \) by a one hidden-layer neural network \( N \), that is:

\[
|f(x) - N(x)| \leq \epsilon, \quad \forall x
\]

(5.0)

Lemma 2. (Bartlett and Mendelason 2002) Let \( \mathcal{F} \) and \( \mathcal{G} \) be two hypothesis classes and let \( a \in \mathbb{R} \) be a constant, we have:

\[
\mathcal{F} \subseteq \mathcal{G} \Rightarrow \hat{R}^a_S(\mathcal{F}) \leq \hat{R}^a_S(\mathcal{G}) 
\]

(6.0)

Lemma 3. (Mohri, Rostamizadeh, and Talwalkar 2012) Let \( z \) be a random variable of support \( Z \) and distribution \( D \). Let \( S = \{ z_1, z_2, \ldots, z_N \} \) be a data set of \( N \) i.i.d. samples drawn from \( D \). Let \( \mathcal{F} \) be a hypothesis class satisfying \( \mathcal{F} \subseteq \{ f : f : Z \to [a, a + 1] \} \). Fix \( \delta \in (0, 1) \). With probability at least \( 1 - \delta \) over the choice of \( S \), the following holds for all \( h \in \mathcal{F} \):

\[
\mathbb{E}_D[h] \leq \hat{E}_S[h] + 2\hat{R}^a_S(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{2N}} 
\]

(7.0)

Definition 1 (Parametric Convolutional Kernel Space). Let \( f \) be a function that maps vector from \( \mathbb{R}^n \) to matrix in \( \mathbb{R}^{n \times m} \), \( n, m \in \mathbb{N}^+ \), and we call this function as convolution kernel generator function. Then we define Parametric Convolutional Kernel Space \( \mathcal{K}_f \) as:

\[
\mathcal{K}_f = \{ [f(p^1), f(p^2), \ldots, f(p^{od})] | p^i \in D \subset \mathbb{R}^n, \quad i = 1, 2, \ldots, od \}. 
\]

(8.0)

We call \( n \) the parameter number, \( m \) the kernel size, and \( od \) (short for output dimension) the output dimension. Since a convolutional kernel in a parametric convolutional kernel spaces is generated by function \( f \), we call \( f \) as the generator function, and \( f_{i,j}(p) = f(p)[i,j] \) as the pixel generator function.

Once we have defined the parametric convolutional kernel space, we can use \( n \) parameters to generate \( m \times m \) sized convolution kernel through the generator function \( f \), which means that the parameters in a convolutional layer can be reduced if \( n \ll m \times m \). However, the reduction in parameters often causes loss of performance as the hypothesis space becomes smaller. Therefore, how the reduction of parameters affects the performance is the key point. We want to study the simplest situation, that is to say, we want to replace the ordinary kernel in the first convolutional layer with the parameter kernel generated from a parametric convolutional kernel space.

Definition 2 (Geometric Operator CNN). Assume that \( \mathcal{K}_f \) is a parametric convolutional kernel space. If the kernel in the first convolutional layer of a convolutional neural network is generated from \( \mathcal{K}_f \), we call this network Geometric Operator CNN. We denote the set of Geometric Operator CNN by \( \mathcal{G}_f \).

Geometric Operator CNN is almost exactly the same as common CNN, except for the kernel in the first convolutional layer. We treat the first convolutional layer as a function from images to outputs, which then act as input of the following layer. If this function is not an injective function, meaning that different inputs can be mapped to identical outputs, then the network takes these identical outputs as the input of the following layers, meaning that the final outputs are still the same. However, the image inputs of the first convolutional layer are different, and corresponding labels can also be different. Thus, when the final outputs are the same, errors must occur.

Therefore, we need to choose kernel carefully to make the function be an injective function. Since the convolution operator is a linear operator, we have the following proposition.

Proposition 1. If the kernel of a convolutional layer, denoted by \( w \), satisfies the following:

\[
I * w = 0 \iff I = 0, \quad \forall I
\]

(9.0)

where \( I \) is the layer input and \( * \) is the convolution operation, then this convolutional layer is an injective function.

We find a necessary and sufficient condition for a convolutional layer to be an injective function. But which kernel satisfies this condition? In the proposition below, we show that \( 3 \times 3 \) kernel generated by Gabor filter function satisfies this condition.

Proposition 2. Let \( f \) be the Gabor filter function, that is \( f_{x,y}(\theta, \sigma, \gamma, \lambda, \psi) = \exp(-x'^2 + y'^2) \cos(2\pi \frac{x'}{\lambda} + \psi) \), where \( x' = x \cos\theta + y \sin\theta, y' = -x \sin\theta + y \cos\theta \). Let \( \mathcal{K}_f \) be the corresponding parametric convolutional kernel space with kernel size \( m \) equal to 3 and sufficient output dimension \( od \). Then, there exists kernel in \( \mathcal{K}_f \) satisfies the condition (9.0).

As the kernel generated from \( \mathcal{K}_f \) could not meet the (9.0), we have the following definition:

Definition 3 (Well-Defined Geometric Operator CNN). Let \( G \in \mathcal{G}_f \), if there is a kernel generated by \( \mathcal{K}_f \) that satisfies (9.0), we call \( G \) a well-defined Geometric Operator CNN.
CNN. We denote the set of all well-defined Geometric Operator CNNS as $G_f$.

**Corollary 1.** If the generator function $f$ is Gabor filter function, the Geometric Operator CNN is well-defined.

Now, let us consider a Convolutional Neural Network with one convolutional layer and two fully-connected layers, and we will study the convergence of common CNN and Geometric Operator CNN. For the common CNN, denoted by $F$, we define the convolution kernel as $k_F$. The weights of the rest of the fully-connected layers are $\{a_F, 0, a_F, 1, b_F, 2\}$, and the biases of three layers are $\{b_{F, 0}, b_{F, 1}, b_{F, 2}\}$. Let $\sigma$ stand for sigmoid activation function, then the convolutional layer $C_F$ and the fully-connected layer $FC_F$ can be defined as follows:

$$C_F(x) = x \ast k_F + b_{F, 0}$$

$$FC_{F, k}(x) = a_{F, k}x + b_{F, k}, \quad k = 1, 2 \quad (10.1)$$

Then, the last two fully-connected layers can be defined as:

$$D_F(x) = FC_{F, 2} \circ \sigma \circ FC_{F, 1}(x) \quad (10.2)$$

Therefore, the output before activation, denoted by $F(x)$, and after activation, denoted by $\tilde{F}(x)$, are defined as:

$$F(x) = D_F \circ \sigma \circ C_F(x)$$

$$\tilde{F}(x) = \sigma \circ F(x) \quad (10.3)$$

We denote the set of common CNN as $F$, that is, $F = \{F\}$, and the output before activation and after activation of input $I_i$ as $F_i, \tilde{F}_i$.

For a Geometric Operator CNN $G$, we similarly define the convolutional kernel to be $k_G$, and the weights and biases are $\{a_{G, 1}, a_{G, 2}, b_{G, 0}, b_{G, 1}, b_{G, 2}\}$. Then, we have the following shorthand when the input is $x$:

$$C_G(x) = x \ast k_G + b_{G, 0}$$

$$FC_{G, k}(x) = a_{G, k}x + b_{G, k}, \quad k = 1, 2$$

$$D_G(x) = FC_{G, 2} \circ \sigma \circ FC_{G, 1}(x) \quad (10.4)$$

$$G(x) = D_G \circ \sigma \circ C_G(x)$$

$$\tilde{G}(x) = \sigma \circ G(x)$$

We denote the output before activation and after activation of input $I_i$ as $G_i, \tilde{G}_i$ as well.

We maintain the same neuron number for each corresponding layer in common CNN and Geometric Operator CNN, that is to say, $\dim(b_{F, k}) = \dim(b_{G, k}), k = 0, 1, 2$, since the approximation ability is different when the neuron number is different. We define the width of each layer as $d_k = \dim(b_{F, k}) = \dim(b_{G, k}), k = 0, 1, 2$.

Then, the empirical loss function for common CNN and Geometric Operator CNN is:

$$\hat{E}_S[F] = \frac{1}{N} \sum_{i=1}^{N} (\tilde{F}_i - y_i)^2$$

$$\hat{E}_S[G] = \frac{1}{N} \sum_{i=1}^{N} (\tilde{G}_i - y_i)^2 \quad (10.5)$$

We have the following theorem on the difference of these two loss functions.

**Theorem 1.** For any $F \in F$, where $F$ is the set of common CNN, if the first fully-connected layer is wide enough, the empirical loss of a well-defined Geometric Operator CNN can be that of common CNN controls. That is, for an arbitrary $\epsilon > 0$, there exists $d^* \in \mathbb{N}^+$ and $G \in G_f$, such that when $d_1 \geq d^*$, the following inequality holds:

$$|\hat{E}_S[G] - \hat{E}_S[F]| \leq \epsilon \quad (11.0)$$

**Theorem 2.** For any $F \in F$, where $F$ is the set of common CNN, if the first fully-connected layer is wide enough, the generalization error of a well-defined Geometric Operator CNN can be that of common Convolutional Neural Network controlled. That is, for an arbitrary $\epsilon > 0$, there exists $d^* \in \mathbb{N}^+$ and $G \in G_f$, such that when $d_1 \geq d^*$, the following inequality holds:

$$\hat{E}_D[G] \leq \hat{E}_D[F] + 2\mathfrak{K}_F^2(F) + \sqrt{\log(1/\delta)/2N} + \epsilon \quad (12.0)$$

In Theorem 2, we know that well defined Geometric Operator CNNs have almost the same generalization error as common CNNs. Therefore, we need to find which Geometric Operator CNNs are well defined.

As Geometric Operator CNN with Gabor filter function as the generator function is well defined, we have the following corollary.

**Corollary 2.** Let $f$ be Gabor filter function, for any $F \in F$, if the first fully-connected layer is wide enough, the generalization error Geometric Operator CNN $G$, which applies $f$ as the generator function can be that of $F$ controlled. That is, for an arbitrary $\epsilon > 0$, there exists $d^* \in \mathbb{N}^+$ and $G \in G_f$, such that when $d_1 \geq d^*$, the following inequality holds:

$$\hat{E}_D[G] \leq \hat{E}_D[F] + 2\mathfrak{K}_F^2(F) + \sqrt{\log(1/\delta)/2N} + \epsilon \quad (13.0)$$

More generally, if there are many generator functions in the first convolutional layer of a Geometric Operator CNN, when the number of kernels generated by Gabor filter function is sufficient enough, this Geometric Operator CNN is also well defined. Therefore, we have the following corollary.

**Corollary 3.** Let $\{f_1, f_2, \ldots, f_T\}$ be the set of generator functions. Suppose that there are $d$ convolution kernels
\{k_1, k_2, \ldots, k_{\text{od}}\}\) in the first convolutional layer of a Geometric Operator CNN, denoted by \(G\), and each \(k_j\) is generated by function \(f_{t_j}\), where \(1 \leq j \leq \text{od}\), \(1 \leq t_j \leq T\). If there exists \(t^* \in \{1, 2, \ldots, T\}\) such that \(f_{t^*}\) is Gabor filter function, and the number of kernels generated by \(f_{t^*}\), denoted by \(n_{t^*}\), is sufficient big enough, then \(G\) is well defined, so that (12.0) holds.

5 Experiments

In the previous chapter, we give theoretical assurance for the Geometric Operator Convolutional Neural Network. The following section includes an explanation of the experiments conducted on the geometric Operator Convolutional Neural Network. All experiments are performed on a single machine with CPU Intel Core i7-7700 CPU @ 3.60GHz 8, GPU TITAN X (Pascal), and RAM 32G.

5.1 Approximation accuracy, generalization error, and feature visualization

Approximation accuracy and generalization error Theoretical analyses ensure that the Geometric Operator Convolutional Neural Network has the same approximation accuracy and the same upper bound for generalization error as the common Convolutional Neural Network. We verify this using two kinds of experiments on CIFAR-10/100. The generalization error refers to the performance of the model on the test set, and the approximation accuracy refers to the performance of the model on the training set.

Figure 5: CIFAR-10

Recognizing objects in an actual scene is not dependent on corresponding domain knowledge but on humans’ prior information. For object recognition tasks, the Geometric Operator Convolutional Neural Network’s recognition effect is worth exploring. The commonly used public data sets for common object recognition are CIFAR-10/100 categories. They are all three-channel color images with a resolution of 32×32. The train set contains 50,000 images and the test set contains 10,000 images. As shown in Fig. 6, ResNet18, ResNet34, and ResNet50 were used on these two public datasets. In the experiment, four paddings were added on the four edges. Then, a random 32×32 cropping was performed, and a data enhancement method was carried out, which involved turning the image up and down. For both testing and training, the images’ pixels are normalized to a 0-1 distribution. The Stochastic gradient descent optimization algorithm with 0.9 the momentum (Loshchilov and Hutter 2016) was used during the training process. The batch size was 100, the initial learning rate was 0.1, and the weight decay was 0.0005. The learning rate was reduced by one fifth per 60, 120, and 160 epochs. We report the performance of our algorithm on a test set after 200 epochs based on the average over five runs.

As shown in Fig. 7, according to the cross-entropy curve of the CIFAR-10 and CIFAR-100 train sets, GO-CNN’s value initially fell faster than the common CNN’s, eventually almost reaching the same value. It is verified that Geometric Operator Convolutional Neural Network achieves the same approximation accuracy as the common Convolutional Neural Network. According to the error rate curve of the CIFAR-10 and CIFAR-100 verification set (Fig. 8), the value of Geometric Operator Convolutional Neural Network is lower than that of the common Convolutional Neural Network. In addition, as shown in Tab. 1, the Geometric Operator Convolutional Neural Network on the CIFAR-10 test

Table 1: The model’s accuracy rates averaged over five experiments on the test set
Figure 7: Log of cross entropy curve during training in common ResNet 18-34-50 and GO-ResNet 18-34-50.

set was 0.4% more accurate than the common Convolutional Neural Network. On the CIFAR-100 test set, the GO-CNN was 0.5% more accurate than the common CNN. It is verified that Geometric Operator Convolutional Neural Network achieves the same generalization error bound as the common Convolutional Neural Network.

**Feature visualization** One way to evaluate a model is through visualizing the features that the model extracts; this is called feature visualization. T-SNE (Maaten and Hinton 2008) or PCA (Jolliffe 2011) are generally used for visualization. The T-SNE visualization maps data points to a two-dimensional or three-dimensional probability distribution through affinity transformation. Then, the data points are displayed with a two-dimensional or three-dimensional plane.

In this paper, a two-dimensional T-SNE visualization is adopted to display the CIFAR-10 features extracted by the model. As shown in Fig. 9, the CIFAR-10 features extracted by the Geometric Operator Convolutional Neural Network are evenly separated from each other in the two-dimensional visualization of T-SNE, while the features extracted from the common Convolutional Neural Network are mixed. It is apparent that the features extracted by the Geometric Operator Convolutional Neural Network are more separable; in other words, the features learned by the Geometric Operator Convolutional Neural Network are more distinguishable and easy to classify with the last fully connected layer.

The numerical experimental results and the feature visualizations of the two datasets reveal that the Geometric Operator Convolutional Neural Network achieves the same approximation accuracy and the same upper bound for the generalization error as the common Convolutional Neural Network. Moreover, the features extracted by Geometric Operator Convolutional Neural Network are more distinguishable.

### 5.2 Generalization

In many practical applications, such as the military, medical care, and so on, annotated data are often insufficient. Thus, a model’s generalization ability for small data sets is of great importance. The generalization ability refers to the ability of a model to predict unknown data when it has been learned by a certain method.

For the open datasets CIFAR-10/100 and MNIST, their train sets are large and their test sets are small. MNIST is a public, handwritten recognition dataset with a total of ten classes. This dataset is shown in Fig. 10 as a channel image.
Figure 8: Error rate curve during training in common ResNet 18-34-50 and GO-ResNet 18-34-50.

Figure 9: T-SNE two-dimensional visualization of CIFAR-10

with 28×28 resolution and a clean background. There are 50,000 train sets, 5,000 verification sets, and 10,000 testing sets. In these numerical experiments, the test set is directly used to train the model, and the train set is used to evaluate the model. These experiments assess the generalization ability of the Geometric Operator Convolutional Neural Network and the common Convolutional Neural Network.

Many training techniques have been used in numerical experiments with CIFAR and MNIST. For numerical experiments with the CIFAR-10/100, the techniques and models used are the same as in Sec. 5.1. For numerical experiments with the MNIST data set, the adaptive moment estimation (Adam (Kingma and Ba 2014)) optimization algorithm was used. In addition, as an image enhancement strategy, the image padding was increased to 32×32 during the training process. The batch size was set to 11, the initial learning rate was 0.001, and the weight decay was 0.0005. The learning rate stays the same until reaching 20,000 iterations. Consequently, we complete 20,000 iterations on one test set and average the performance over five runs in order to report the final performance evaluation of our algorithm. The basic network structure used in the experiment is LeNet (LeCun et al. 1998) as shown in Fig. 11. There are two convolution layers and two fully-connection layers in the network. Similarly, in the Geometric Operator Convolutional Neural Network, the first convolutional layer is replaced by the operator convolutional layer. The convolution kernels from the first layer are composed of trainable Gabor kernels and Schmid kernels. The other convolutional layers are the common convolutional layers.

As shown in Tab. 2, from the perspective of the accuracy of MNIST and CIFAR-10/100, after the train set drops
Table 2: The accuracy of the test sets for the small train set and the large train set (in brackets) as averaged over five experiments

| Model       | CIFAR − 10            | CIFAR − 100          | MNIST          |
|-------------|-----------------------|----------------------|----------------|
| common ResNet18 | 84.96%(94.79%)       | 44.97%(77.06%)       | −              |
| GO − ResNet18  | 86.21%(95.17%)       | 47.03%(77.59%)       | −              |
| common ResNet34 | 82.33%(95.27%)       | 44.74%(78.26%)       | −              |
| GO − ResNet34  | 86.36%(95.77%)       | 49.00%(78.72%)       | −              |
| common ResNet50 | 83.86%(94.44%)       | 45.93%(78.45%)       | −              |
| GO − ResNet50  | 85.64%(94.72%)       | 47.09%(79.50%)       | −              |
| common LeNet   | −                     | −                     | 97.75%(99.22%) |
| GO − LeNet    | −                     | −                     | 97.97%(99.24%) |

5.3 Adversarial stability

Although the Geometric Operator Convolutional Neural Network reduces the number of trainable parameters, it enhances adversarial stability. The current machine learning model, including the neural network and other models, is vulnerable to attacks from adversarial samples. In addition, Convolutional Neural Network shows instability under attacks against adversarial samples (Goodfellow, Shlens, and Szegedy 2014). Adversarial samples are produced when an attacker misleads a classifier by slightly disturbing the original sample. It is very important to study the stability of adversarial samples in practice. The false alarm rate of existing intelligent video analysis technology is as much as 30% to 60%, which greatly affects the actual application and deployment. For example, the identification system in Tiananmen Square was also removed due to high false alarm rates.

The geometric operator has its own characteristics, and the Schmid operator has rotation invariance. It is worth exploring whether the Geometric Operator Convolutional Neural Network, which is formed by the Schmid operator, enhances the adversarial stability of the adversarial sample when rotated at a certain angle. Geometric operators use domain knowledge and prior knowledge to extract image features. It is worth investigating the Geometric Operator Convolutional Neural Network’s ability to enhance the adversarial stability of the adversarial samples against noise interference. The stability of the model is measured by the difference between the accuracy of the original test set and the adversarial sample generated by the test set.

The open handwriting recognition data set (MNIST) is the primary dataset used in this experiment. The techniques and models are the same as those used for MNIST in Sec. 5.2. Both models are trained on the MNIST train set. original images, adversarial samples of gaussian interference, and adversarial samples from random rotation were used to evaluate the two models.

It can be seen from Tab. 3 that when the test set is randomly rotated within 90 degrees, the difference of the Geometric Operator Convolutional Neural Network is 1.21% lower than that of the common Convolutional Neural Network. This verifies that the Geometric Operator Convolutional Neural Network enhances the adversarial stability of rotated samples. As can be seen from Tab. 4, when the small Gaussian disturbance (the mean is 0, the standard deviation is 0.3) is applied to the test set, the difference of the Geometric Operator Convolutional Neural Network is 0.6%
lower than that of the common Convolutional Neural Network. This indicates that the Geometric Operator Convolutional Neural Network enhances the adversarial stability of Gaussian disturbance adversarial samples. In sum, the Geometric Operator Convolutional Neural Network enhances the adversarial stability of certain adversarial samples.

### 5.4 Application

Medical images in China are developing rapidly, but specialist doctors are short of resources, and they are mainly concentrated in big cities and big hospitals. Many small and medium-sized cities do not have sufficient diagnostic imaging capacities, so many patients have to go to big cities in order to access better medical resources and obtain better treatment. Similarly, there are few orthopaedic surgeons in China. Fractures often occur in real life due to accidents, such as falls and car accidents. There are many ways to obtain medical data, such as X-ray images, CT images, MRI images, and representational images; however, orthopedists usually use X-ray images to diagnose fractures. With the development of artificial intelligence technology, many scholars use Convolutional Neural Networks to assist doctors in determining whether a bone image reveals a fracture. (Chung et al. 2018).

In (Fig. 12), however, this method, which is preprocessed by geometric operators, can be considered a two-stage method. The parameters of geometric operators are preset by human experience. At this point, it is difficult for the local parameters obtained by the respective optimization to reach the global optimum. Thus, one may consider integrating the pre-processing of geometric operators into the deep network for global parameter learning without prior artificial empirical design parameters. In other words, this would mean using the Geometric Operator Convolution Neural Network proposed in this paper, wherein the convolution kernels from the first layer are all trainable Schmid kernels.

Figure 12: The framework of the two-stage method

Doctors usually judge whether a fracture has occurred based on whether there is a fracture line (texture) in the image. In (Cao et al. 2015), the texture information from the image is used for an auxiliary diagnosis of a fracture. With prior information from the Schmid operator, we do pre-processing by Schmid operators to enhance the texture information from an image. Then, we use the deep Convolutional Neural Network to conduct classification (as shown in Fig. 12). However, this method, which is preprocessed by geometric operators, can be considered a two-stage method. The parameters of geometric operators are preset by human experience. At this point, it is difficult for the local parameters obtained by the respective optimization to reach the global optimum. Thus, one may consider integrating the pre-processing of geometric operators into the deep network for global parameter learning without prior artificial empirical design parameters. In other words, this would mean using the Geometric Operator Convolution Neural Network proposed in this paper, wherein the convolution kernels from the first layer are all trainable Schmid kernels.

Figure 13: Bone images

Around 2,000 samples from X-rays taken at the Hainan Peoples Hospital were used as the data for the three kinds of intelligent fracture diagnosis models. Each sample was manually divided into bone regions, as shown in Fig. 13, with a total of 5,743 bone regions, including 723 bone fracture regions. The above three models are used for numerical experiments. The basic network framework used in the experiment is ResNet50 (He et al. 2016), which mainly consists of a new residual structure unit (Fig. 6). To balance the data during training, the number of fracture patches is increased to 4,016 by rotating the images and changing the background of the images. In the test set, there were 145 fracture patches and 1,004 non-fracture patches. Then, five experiments were conducted to evaluate each model. The stochastic gradient descent optimization algorithm and the finetune strategy were used during the training process, with a batch size of 50. The initial learning rate was 0.001 and the weight decay was 0.0005. The learning rate is reduced by...
Table 5: Experimental results of intelligent diagnosis

|                      | CNN     | two - stage method | GO - CNN |
|----------------------|---------|--------------------|----------|
| fracture recall      | 87.97%  | 88.74%             | 90.95%   |
| non - fracture recall | 96.57%  | 96.17%             | 96.87%   |

one fifth every 4,000 iterations. Each data class is queued, and the data from each batch is averaged out of each data class during training. We report the performance of our algorithm on the test set after 12,000 iterations based on the average over five runs.

According to Tab. 5, the Geometric Operator Convolutional Neural Network is the most accurate. Moreover, the fracture recall of the two-stage method is 0.77% higher than that of the Convolutional Neural Network, indicating that domain knowledge from the field of medicine is important for intelligent diagnosis. The fracture recall of the Geometric Operator Convolutional Neural Network is 2.21% higher than that of the two-stage method, which indicates that the Geometric Operator Convolutional Neural Network does make use of medical knowledge for fracture diagnosis. The integration of geometric operator into the deep neural network indeed achieve global optimization.

In the above experiments, the Geometric Operator Convolutional Neural Network uses a priori knowledge from the field of medicine and provides a better recognition effect. Although the trainable parameters decrease, GO-CNN still reaches the same approximation accuracy and a slightly lower generalization error upper bound when compared with the common CNN. The features extracted from the Geometric Operator Convolutional Neural Network are more distinguishable, and the Geometric Operator Convolutional Neural Network reduces the dependence on training samples and enhances the adversarial stability of certain adversarial samples. Moreover, the GO-CNN also uses medical knowledge for the practical purpose of assisting in intelligent medical diagnoses of bone fractures.

6 Conclusion and Future Research

In this paper, we present a new framework named the Geometric Operator Convolution Neural Network, where the kernel in the first convolutional layer is replaced with kernels generated by geometric operator functions. This new network boasts several contributions. Firstly, the Geometric Operator Convolution Neural Network is customizable for diverse situations. Other geometric operators may be convolved using the convolution process of the convolution of the Gabor operator and the Schmid operator. Whereas the geometric operator convolution in the GO-CNN can be replaced by different geometric operators, so the GO-CNN is highly versatile. Second, there is a theoretical guarantee in the learning framework of the Geometric Operator Convolutional Neural Network. In this paper, the universal approximation theorem and multiple lemmas are used to prove that GO-CNN reaches the same approximation accuracy and the same generalization error upper bound as the common CNN when certain conditions (i.e., when the training sample is singular) are satisfied. In addition, through experiments on CIFAR-10/100, we verify that GO-CNN reaches the same approximation accuracy with a smaller generalization error when compared to the common Convolutional Neural Network. Thirdly, the Geometric Operator Convolutional Neural Network reduces the dependence on training samples. The train set and the test set of CIFAR-10/100 and MNIST were exchanged and then re-trained and tested. The Geometric Operator Convolutional Neural Network improved the generalization performance by achieving a higher testing accuracy with the same training loss. In other words, GO-CNN has less dependence on the training samples. Lastly, the Geometric Operator Convolutional Neural Network enhances adversarial stability. Gaussian perturbation and random rotation were performed on the MINIST test set and then tested. The experimental results show that the Geometric Operator Convolutional Neural Network enhances adversarial stability. Furthermore, the GO-CNN improves diagnostic efficiency by offering intelligent medical diagnostic assistance based on domain knowledge acquired from images of bone fractures.

In this paper, only the convolutions of two kinds of geometric operators are considered. In the future, we can explore more submodules suitable for the Geometric Operator Convolutional Neural Network, namely, the convolutions of more geometric operators with better performances. We can explore a more appropriate geometric operator convolution block. In addition, we can analyze the internal relations of the Geometric Operator Convolution Network from the theoretical analysis provided in this paper.

7 Acknowledgments

We thank Professor Zhouwang Yang for his technical guidance. Notably, we would like to express our full thanks to Shiwei Wang and Haikou People’s Hospital for providing medical data.

A Appendix

Proof of Proposition1:

Assume that the proposition is not true, then there exist \( I_1 \neq I_2 \), such that \( I_1 \ast w = I_2 \ast w \). Thus, if we set \( I = I_1 - I_2 \), we have \( I \ast w = (I_1 - I_2) \ast w = 0 \), since \( \ast \) is a linear operator, which means that \( I = 0 \) according to the condition. Therefore, the assumption is not true, and the conclusion is proved.

Proof of Proposition2:

Assume that there exists \( I \in \mathbb{R}^{3\times3} \), \( I \neq 0 \), such that \( I \ast k = 0 \) holds for \( \forall k \in K_f \).

We write \( I \) in the following matrix way:
\[ I = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \]  \quad (14.1)

We define the pixel generator function \( f_{ij} \) to be \( f_{x-1,y-1}(\theta, \sigma, \gamma, \lambda, \psi) \). Then, we have the following equivalence:

\[ I \ast k = 0 \iff \sum_{i=0}^{2} \sum_{j=0}^{2} a_{ij} f_{ij} = 0. \]  \quad (14.2)

We will choose a variety of different parameters to discuss.

**I.** \( \theta = 0, \lambda = 1, \psi = 0. \)

Since \( \theta = 0 \), we have \( x' = x, y' = y \), and the following:

\[
\left\{ \begin{array}{l}
    f_{00} = f_{02} = f_{20} = f_{22} = \exp \left( -\frac{1 + \gamma^2}{2\sigma^2} \right) \\
    f_{01} = f_{21} = \exp \left( -\frac{1}{2\sigma^2} \right) \\
    f_{10} = f_{12} = \exp \left( -\frac{\gamma^2}{2\sigma^2} \right) \\
    f_{11} = 1
\end{array} \right. \]  \quad (14.3)

We make the following shorthands for convenience:

\[ h_1 = \exp \left( -\frac{1 + \gamma^2}{2\sigma^2} \right) \]
\[ h_2 = \exp \left( -\frac{1}{2\sigma^2} \right) \]  \quad (14.4)
\[ h_3 = \exp \left( -\frac{\gamma^2}{2\sigma^2} \right) \]

From Eqn.14.2, we can get:

\[(a_{00} + a_{02} + a_{20} + a_{22})h_1 + (a_{01} + a_{21})h_2 + (a_{10} + a_{12})h_3 + a_{11} = 0. \]  \quad (14.5)

The equation above means that, \( \exists b_0, b_1, b_2, b_3 \in \mathbb{R}, \) such that

\[ b_0 + b_1 h_1 + b_2 h_2 + b_3 h_3 = 0. \]  \quad (14.6)

Differentiate on both sides of parameter \( \gamma \) and get:

\[ -\frac{\gamma}{\sigma^2} b_1 h_1 - \frac{\gamma}{\sigma^2} b_3 h_3 = 0 \]

\[ \implies b_1 + \exp \left( \frac{\gamma^2}{2\sigma^2} \right) b_3 = 0 \]  \quad (14.7)

Since Eqn.14.7 holds for \( \forall \gamma, \sigma \), which indicates that:

\[ b_1 = b_3 = 0 \]  \quad (14.8)

In the same way, we can get the following equation from Eqn.14.8:

\[ b_2 h_2 + b_0 = 0 \]
\[ b_0 = b_2 = 0 \]  \quad (14.9)

Therefore, we have the following equations:

\[
\left\{ \begin{array}{l}
    a_{00} + a_{02} + a_{20} + a_{22} = 0 \\
    a_{01} + a_{21} = 0 \\
    a_{10} + a_{12} = 0 \\
    a_{11} = 0
\end{array} \right. \]  \quad (14.10)

**II.** \( \theta = 0, \lambda = 3, \psi = \pi/3. \)

In the same way, we have the following equations:

\[
\left\{ \begin{array}{l}
    f_{20} = f_{21} = f_{22} = 0 \\
    f_{00} = f_{02} = \frac{1}{2} h_1 \\
    f_{10} = f_{12} = \frac{1}{2} h_3 \\
    f_{01} = \frac{1}{2} h_2 \\
    f_{11} = \frac{1}{2}
\end{array} \right. \]  \quad (14.11)

And we can get:

\[(a_{00} + a_{02})h_1 + a_{01} h_2 + (a_{10} + a_{12})h_3 + a_{11} = 0, \]  \quad (14.12)

which indicates that:

\[
\left\{ \begin{array}{l}
    a_{00} + a_{02} = 0 \\
    a_{01} = 0
\end{array} \right. \]  \quad (14.13)

From Eqn.14.10, we can get:

\[
\left\{ \begin{array}{l}
    a_{00} + a_{02} = 0 \\
    a_{01} = a_{21} = 0
\end{array} \right. \]  \quad (14.14)

**III.** \( \theta = 0, \lambda = 3, \psi = -\pi/3. \)

We can get the following equations in the way just the same as discussed in situation II:

\[ a_{20} + a_{22} = 0 \]  \quad (14.15)

**IV.** \( \theta = \pi/2, \lambda = 3, \psi = \pm \pi/3. \)

We have \( x' = y, y' = x \) this time, and we can get the following equations as the way discussed in situation II III, :

\[
\left\{ \begin{array}{l}
    a_{00} + a_{20} = 0 \\
    a_{02} + a_{22} = 0 \\
    a_{10} = a_{12} = 0
\end{array} \right. \]  \quad (14.16)

Combine equations 14.14 14.15 14.16, we can get:

\[
\left\{ \begin{array}{l}
    a_{00} = -a_{02} = a_{22} = -a_{20} = v \\
    a_{01} = a_{10} = a_{12} = a_{21} = 0
\end{array} \right. \]  \quad (14.17)

**V.** \( \theta = \pi/4, \lambda = 2, \psi = \sqrt{2}\pi. \)

We have \( x' = \sqrt{2}(x + y), y' = \sqrt{2}(y - x) \) and the following:

\[
\left\{ \begin{array}{l}
    f_{00} = \exp \left( -\frac{1}{\sigma^2} \right) \\
    f_{02} = f_{20} = \exp \left( -\frac{\gamma^2}{\sigma^2} \right) \\
    f_{22} = \exp \left( -\frac{1}{\sigma^2} \right) \cos(2\sqrt{2}\pi)
\end{array} \right. \]  \quad (14.18)

Therefore, we have
(1 + \cos(2\sqrt{\pi})) \exp(-\frac{1}{\sigma^2})v + 2\cos(2\sqrt{2\pi}) \exp(-\frac{\gamma^2}{\sigma^2})v = 0
\implies v = 0
\implies a_{00} = a_{02} = a_{20} = a_{22} = 0
(14.19)

Combine equations 14.10 14.17 14.19, we can find that
\(a_{ij} = 0, \text{for} i, j = 0, 1, 2\), which means that \(I = 0\). Therefore, the assumption that \(I \neq 0\) is not true.

For an arbitrary sized input \(I\), we can focus on the \(3 \times 3\) sized submatrix that will do inner product with the convolution kernel and get the same conclusion.

**Proof of Corollary 1:**
From Prop. 2, the conclusion is obvious.

**Proof of Theorem 1:**
Notice that
\[
N(\tilde{E}_S[G] - \hat{E}_S[F]) = \sum_i \tilde{G}_i^2 - \tilde{F}_i^2 - 2y_i(\tilde{G}_i - \tilde{F}_i)
\]
\[
= \sum_i (\tilde{G}_i - \tilde{F}_i)(\tilde{G}_i + \tilde{F}_i - 2y_i)
(15.1)
\]
Apply absolute value on both sides
\[
N[|\tilde{E}_S[G] - \hat{E}_S[F]| \leq \sum_i |\tilde{G}_i - \tilde{F}_i||\tilde{G}_i + \tilde{F}_i - 2y_i|
\]
\[
\leq 4 \sum_i |\tilde{G}_i - \tilde{F}_i|
(15.2)
\]
The last inequality holds as \(|\tilde{F}_i|, |\tilde{G}_i|, |y_i| \leq 1\).
We can find parameters of ordinary CNN, so that there is a mapping between input \(I_i\) and output \(\tilde{F}_i\), and the mapping function is \(\tilde{F}\) as we have defined.

We can also fix parameters of \(C_G\), and choose the convolution kernel of \(C_G\) that satisfies (9.0) since \(G\) is a well-defined Geometric Operator CNN, so that \(C_G\) is an injective function, which means that \(C_G\) exists. In the same time, \(D_G = FC_{G,2} \circ \sigma \circ FC_{G,1}\) can be treated as a one hidden layer neural network.

Define a new hypothesis \(\tilde{h} = \sigma \circ F \circ C_G^{-1} \circ \sigma^{-1}\) ranges in \([0, 1]\), according to Lemma. 1, we can find parameters \(\{a_{G,k}, b_{G,k}\}, k = 1, 2\), such that
\[
|\sigma \circ D_G(x) - \tilde{h}(x)| \leq \epsilon / 4, \quad \forall x.
(15.3)
\]
Replace \(x\) by \(\sigma \circ C_G(I_i)\) we can get
\[
|\tilde{G}_i - \tilde{F}_i|
= |\tilde{G}(I_i) - \tilde{F}(I_i)|
= |\sigma \circ D_G \circ \sigma \circ C_G(I_i) - \sigma \circ F(I_i)|
\]
\[
= |\sigma \circ D_G(\sigma \circ C_G(I_i)) - \sigma \circ F \circ C_G^{-1} \circ \sigma^{-1}(\sigma \circ C_G(I_i))| \leq \epsilon / 4
(15.4)
\]
Combine with (15.2), we can get
\[
|\hat{E}_S[G] - \hat{E}_S[F]| \leq 4 \sum_i |\tilde{G}_i - \tilde{F}_i| \leq \frac{4N}{4} \epsilon = \epsilon
(15.5)
\]

**Proof of Theorem 2:**
From Theorem 1, we know that \(G\) satisfies the following inequality:
\[
|\hat{E}_S[G] - \hat{E}_S[F]| \leq \epsilon
(16.1)
\]
From Lemma 3, we know that
\[
\hat{E}_D[G] \leq \hat{E}_S[G] + 2\tilde{N}_S(G_f^*) + \sqrt{\frac{\log(1/\delta)}{2N}}
\]
\[
\leq \hat{E}_S[F] + 2\tilde{N}_S(F) + \sqrt{\frac{\log(1/\delta)}{2N}} + \epsilon
(16.2)
\]
Since \(G_f^* \subset F\), we have the following inequality from Lemma 2:
\[
\tilde{N}_S(G_f^*) \leq \tilde{N}_S(F)
(16.3)
\]
Combined with (16.2), we have
\[
\hat{E}_D[G] \leq \hat{E}_S[F] + 2\tilde{N}_S(F) + \sqrt{\frac{\log(1/\delta)}{2N}} + \epsilon
(16.4)
\]
The conclusion is proved!

**Proof of Corollary 2:**
From Theorem 2 and Corollary 1, this conclusion is obvious.

**Proof of Corollary 3:**
Let \(K\) be the set of \(k_i\) such that the generator function of \(k_i\) is \(f_i = f\), and denote the concatenation of all these \(k_i\) as \(\tilde{k}\).

Suppose that there exists an input \(x\), satisfies that \(x + k_i = 0, i = 1, 2, \ldots, od\), then \(x + \tilde{k} = 0, \forall \tilde{k} \in K\). Therefore, \(x + \tilde{k}\) holds for any parameters. However, it is conflict with Prop. 2.
Therefore, the conclusion is proved!

**References**

[Bartlett and Mendelson 2002] Bartlett, P. L., and Mendelson, S. 2002. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* 3(Nov):463–482.

[Calderón, Roa, and Victorino 2003] Calderón, A.; Roa, S.; and Victorino, J. 2003. Handwritten digit recognition using convolutional neural networks and gabor filters. *Proc. Int. Congr. Comput. Intell.*

[Cao et al. 2015] Cao, Y.; Wang, H.; Moradi, M.; Prasanna, P.; and Syeda-Mahmood, T. F. 2015. Fracture detection in x-ray images through stacked random forests feature fusion. In *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*, 801–805. IEEE.
2001 IEEE Computer Society Conference on, volume 2, II–II. IEEE.

[van Vliet, Young, and Beckers 1989] van Vliet, L. J.; Young, I. T.; and Beckers, G. L. 1989. A nonlinear laplace operator as edge detector in noisy images. *Computer vision, graphics, and image processing* 45(2):167–195.

[Wang et al. 2018] Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Works] Works, H. I. Sobel edge detector. [cse. secs. oakland. edu](http://cse. secs. oakland. edu).

[Yao et al. 2016] Yao, H.; Chuyi, L.; Dan, H.; and Weiyu, Y. 2016. Gabor feature based convolutional neural network for object recognition in natural scene. In *Information Science and Control Engineering (ICISCE), 2016 3rd International Conference on*, 386–390. IEEE.