Towards Diffeomorphism Invariant Convolution Neural Networks

Pengyu Kan
UW-Madison
pengyu@cs.wisc.edu

Rudrasis Chakraborty
UC Berkeley
rudra@berkeley.edu

Vishnu Suresh Lokhande
UW-Madison
lokhande@cs.wisc.edu

Vikas Singh
UW-Madison
vsingh@biostat.wisc.edu

Abstract

Most deep neural networks are not natively invariant or equivariant to transformations of the data, e.g., when data samples transformed by a group action, the prediction of the model may change in arbitrary ways. Often, one resorts to data augmentation to tackle this problem. A number of recent ideas show how the network can be modified and endowed with such capabilities, that are variously useful in many applications including robustness. This route avoids the need for explicit data augmentation. In this work, we explore whether performs convolution in the jacobian space of images can offer benefits. We propose DiffCNN, a model which mitigates the performance drop many networks often face when presented with warped versions of the images they were trained on. Our preliminary results show that DiffCNN can minimize the need for data augmentation as well as offers robustness on CIFAR10 dataset relative several alternatives from the literature.

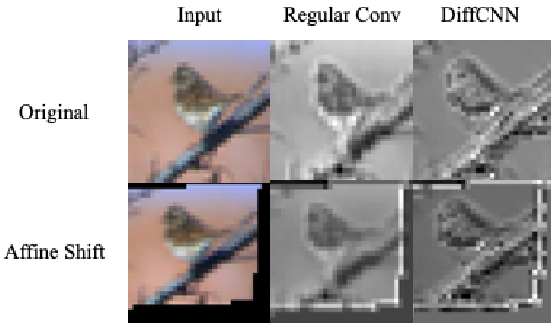

Figure 1. Illustration of latent invariant representation of an image from CIFAR10. Here, we used the output from the convolution layer of a trained CNN model (the base classification architecture from the Section 3), and the invariant output from the trained version of our proposed DiffCNN model. The first row shows the original input and the corresponding generated representations. The second row shows the spatial deformed input under the affine shift transformation. Both model classifies the original input correctly as a bird while only the proposed DiffCNN classifies the deformed input correctly.

1. Introduction

Convolution Neural Networks (CNN) are the default architecture of choice in in many computer vision problems [12, 13, 14]. However, it is known that a standard CNN is not robust to spatial transformations or group actions applied to the input images, and the classification accuracy on such "transformed" data can be significantly worse compared to the model's performance on the original (undeformed) images. For example, in Figure 1, a fully trained CNN model, with seven hidden convolution layers (the base classification architecture in the Section 3), predicts the original image correctly as a bird. In contrast, the model predicts the deformed image as a "plane" instead. Here, the deformed image is generated simply by applying an affine shift transformation on the input image. We find that depending on the amount of deformation/transformation added, the prediction accuracy of a standard CNN on CIFAR10 drops from 85% to 34%.

Motivated by the need to better understand generalization and robustness of CNN model, there is growing interest in understanding the symmetry in the latent representations of the architecture, specifically with respect to invariance or equivariance of the representations to the action of a group [1]. Translation transformations on the input images are handled within CNNs using pooling layers [6]. However, CNNs are not equivariant/invariant to other types of transformations which has been a focus of recent work [2, 6, 10]. In the present paper, our interest is in the broader class of *diffeomorphism* operations which has also been independently studied via augmentation [7] and wavelet-like filter strategies [4].

This paper. We investigate potential benefits of operat-

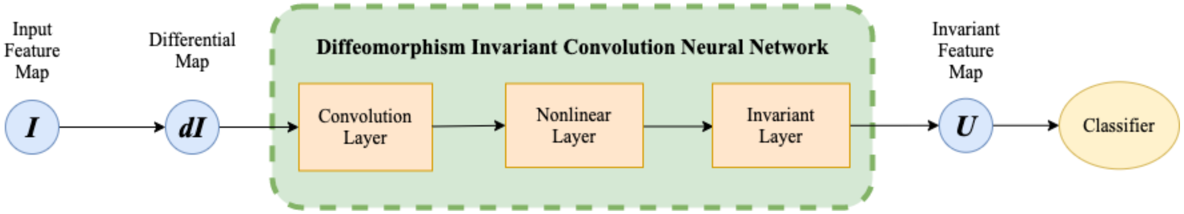

Figure 2. Architecture diagram for Diffeomorphism Invariant Convolution Neural Network (DiffCNN). First, the images are projected into the Jacobian space. Then, a differential map of the images are provided as input to DiffCNN which generates diffeomorphism invariant representations. The detail for the components of the DiffCNN is in Section 2. The invariant layer will project data back into the image space and form the invariant feature map. Lastly, the invariant feature map is fed into a regular classifier.

ing in the Jacobian space of images. A convolution structure on this representation turns out to be invariant to diffeomorphisms under some conditions. We call this convolution layer diffeomorphism invariant convolution neural network (DiffCNN). Our module can be simply inserted into an existing classifier such as the regular CNN and the *Residual Neural Network* (ResNet). In Section 2, we provide the theoretical setup and proof of the diffeomorphism invariance of DiffCNN. Empirically, we achieve the state-of-the-art classification accuracy on affine and homography deformed images from the CIFAR10 [11] dataset, with comparing to the results of several state-of-the-art baseline models in Section 3.

1.1. Related Work

To allow the CNN model to deal with spatial transformations of the images, Jaderberg *et al*. designed a *Spatial Transformer* (ST) for CNN, which is inserted into a regular CNN and enables it to spatially manipulate the input images inside the network [8]. Detlefsen *et al*. proposed using the CPAB transformations within the ST layer which are based on the integration of Continuous Piecewise-Affine (CPA) velocity fields [7]. However, ST layers rely on prior information on all potential spatial transformations of test time data one may encounter. Recent literature proposed applying steerable filters [16], considering the continuous rotational group $SO(2)$ [15], or applying the 2D-discrete-Fourier transform (2D-DFT) [5] to achieve rotational equivariance. In order to reduce the number of training parameters and computational complexity of the rotational equivariant CNN, Cheng *et al*. proposed decomposing the convolution kernels over the joint feature space and designed a decomposition of convolutional filters called RotDCF [4]. However, many of these works restrict the treatment to a specific set of transformations, e.g., translation and rotation.

2. Diffeomorphism Invariant CNN (DiffCNN)

As we see in Figure 2, the DiffCNN will first generate a representation (which is more favorable for diffeomorphism invariance) for the input image (Figure 1), and this generated representation is fed into the subsequent classifier. In this section, we will define the structure of our proposed DiffCNN, and show some properties regarding invariance to diffeomorphism.

2.1. Convolution Layer

We can identify an image $I : \mathbf{R}^2 \to \mathbf{R}^3$ by its corresponding differential form (field of Jacobian) and denote it by $dI : \mathbf{R}^2 \to \mathsf{GL}(3,2)$. We can compute $dI(i,j)$ by computing the Jacobian over a small neighborhood (say, 3×3) around pixel (i,j). Now, we can define our convolution operator as follows. Given $dI = [J_{ij}]_{i=1,j=1}^{N,N} \subset \mathsf{GL}(3,2)$ and a learnable kernel $W = [w_{ij}]_{i=1,j=1}^{K,K} \subset \mathsf{GL}(3,3)$, our convolution operator is

$$(dI \star W)(i,j) = \sum_{k=1,l=1}^{K,K} w_{k,l} J_{i+k-1,j+l-1} \tag{1}$$

The scaling parameters, obtained via singular value decomposition, are shared across all W's. Thus, there are three scaling parameters for W, and six parameters corresponding to the rotation angles for each w_{ij}. As a result, when using a $K \times K$ kernel, this will need a total of $3 + K \times K \times 6$ parameters.

Remark 1. *Introducing additional constraints on W may further reduce the number of parameters. An example of such a constraint is to assume w_{ij} to be a symmetric matrix. This assumption brings down the parameter count to $3 + K \times K \times 3$.*

Diffeomorphisms $G = \{\phi : \mathbf{R}^2 \to \mathbf{R}^2\}$ act on images in the following way: given $\phi \in G$, the action of the diffeomorphism acts on I is as $I \circ \phi$. The group $H = \{d\phi : \mathbf{R}^2 \to \mathbf{R}^2\}$ acts on the space of $\{dI\}$. Here, $d\phi$ is a linear map and can be identified by 2×2 full rank matrices, i.e., we consider $H = \{d\phi : \mathbf{R}^2 \to \mathsf{GL}(2,2)\}$. Define a subgroup of H, denoted by $H_{K,K}$, that consists of all $d\phi$ such that, $d\phi(x_1,y_1) = d\phi(x_2,y_2)$ if $|x_1 - x_2| \leq K$ and $|y_1 - y_2| \leq K$. We will assume non overlapping windows.

Theorem 2. *Under the action of $H_{K,K}$ on $\{dI\}$, the convolution operator in (1) is $H_{k,k}$ equivariant.*

Model	Clean Image	Affine Normal				Affine + Shift				Homography			
		0.01	0.05	0.1	0.5	0.01	0.05	0.1	0.5	0.01	0.05	0.1	0.5
Regular CNN [3]	85%	38%	38%	37%	18%	39%	40%	35%	**14%**	39%	39%	34%	14%
ST + CNN [8]	86%	44%	46%	45%	18%	46%	46%	41%	**14%**	46%	46%	45%	15%
CPAB ST + CNN [7]	85%	47%	47%	43%	17%	48%	47%	40%	13%	49%	45%	37%	13%
RotDCF + CNN [4]	**87%**	**59%**	59%	56%	**21%**	59%	**58%**	**52%**	**14%**	**60%**	**58%**	**52%**	**15%**
DiffCNN + CNN	83%	**59%**	**60%**	**57%**	**21%**	**60%**	**58%**	51%	13%	59%	**58%**	51%	**15%**

Table 1. Classification performance as compared to baseline models on CIFAR10. Our proposed model's result is highlighted with a shaded background. The best accuracy in each test case is marked in **bold**. Checkpoints are selected at the highest validation accuracy of clean images in the 100 epochs. The DiffCNN model shows a similar robustness as the RotDCF + CNN model and DiffCNN's accuracy is around 11% higher than other baseline models on small scale spatial transformation (scale ≤ 0.1).

Proof. Denote the deformed image $I' = I \circ \phi$. Then, we have $dI' = [J'_{ij}]_{i=1,j=1}^{N,N}$, where $J'_{ij} = J_{\phi(i,j)} d\phi(i,j) \approx J_{ij} d\phi(i,j)$. Using J'_{ij} in (1), we get that $(dI' \star W)(i,j)$ is

$$\sum_{k=1,l=1}^{K,K} w_{k,l} J_{i+k-1,j+l-1} d\phi(i+k-1, j+l-1) \tag{2}$$

Based on the assumption of $H_{k,k}$, we have:

$$\begin{aligned}(dI' \star W)(i,j) &= \left(\sum_{k=1,l=1}^{K,K} w_{k,l} J_{i+k-1,j+l-1} \right) d\phi(i,j) \\ &= (dI \star W)(i,j) d\phi(i,j)\end{aligned}$$

■

2.2. Nonlinear Layer

Given the output of the convolution layer to be

$$O = \left[O^k\right] = [J_{ijk}]_{i=1,j=1,k=1}^{N,N,C} \subset \mathsf{GL}(3,2),$$

we define the nonlinear layer as an element-wise operation,

$$\Phi(J_{ijk}) = \begin{cases} J_{ijk} & \text{if } \det(J_{ijk}^T A J_{ijk}) > 0 \\ -J_{ijk} & \text{o.w.} \end{cases} \tag{3}$$

$A \in \mathsf{SO}(3,3)$ is learnable (parameterized with Cayley map).

Corollary 3. *The non-linear operator Φ is equivariant under any group H.*

Proof. Since the previous layer is equivariant to diffeomorphism ϕ, we have $J'_{ijk} = J_{ijk} d\phi(i,j)$. Therefore,

$$\begin{aligned}\det({J'_{ijk}}^T A J'_{ijk}) &= \det(d\phi(i,j)^T J_{ijk}^T A J_{ijk} d\phi(i,j)) \\ &= \det(d\phi(i,j))^2 \cdot \det(J_{ijk}^T A J_{ijk})\end{aligned}$$

Observe that the sign of $\det(J_{ijk}^T A J_{ijk})$ is the same as the sign of $\det({J'_{ijk}}^T A J'_{ijk})$. Thus,

$$\Phi(J'_{ijk}) = J'_{ijk} = J_{ijk} d\phi(i,j), \text{ equivariant to } d\phi(i,j)$$

■

2.3. Invariant Layer

Given the output of the nonlinear layer to be $O = \left[O^k\right] = [J_{ijk}]_{i=1,j=1,k=1}^{N,N,C} \subset \mathsf{GL}(3,2)$, we define the following invariant layer operation:

$$V_{ijk} = (J_{ijk} - \bar{J}_{ij})^T (J_{ijk} - \bar{J}_{ij}) \tag{4}$$

Here, $\bar{J}_{ij} = \frac{1}{C} \sum_{k=1}^{C} J_{ijk}$. In order to map $V_{ijk} \in GL(2,2)$ into $\mathbf{R}$, such that the output of this layer is in an image space: $\mathbf{R}^2 \to \mathbf{R}^C$, we calculate the determinant of V_{ijk}. To separate the action $d\phi$ out of the determinant of V_{ijk}, we add a logarithm operation: $g_{ijk} = \log(|\det(V_{ijk})|)$. Then, to eliminate the component from the action $d\phi$, we perform a normalization by subtracting the average for pixel (i,j) across different channels. This gives

$$U_{ijk} = g_{ijk} - \bar{g}_{ij} \ \text{ where } \ \bar{g}_{ij} = \frac{1}{C} \sum_{k=1}^{C} g_{ijk} \tag{5}$$

Corollary 4. *Under any group action of H on $\{dI\}$, the invariant layer is H-invariant.*

Proof. As the convolution and the non-linear layers are equivariant to ϕ, we have $J'_{ijk} = J_{ijk} d\phi(i,j)$. Thus:

$$\bar{J}'_{ij} = \frac{1}{C} \sum_{k=1}^{C} J'_{ijk} = \frac{1}{C} \sum_{k=1}^{C} J_{ijk} d\phi_{i,j} = \bar{J}_{ij} d\phi(i,j)$$

Substituting J'_{ijk} and $\bar{J}'_{ij}$ into (4), we have:

$$V'_{ijk} = d\phi(i,j)^T V_{ijk} d\phi(i,j)$$

Now, we put V'_{ijk} into the formulation of g'_{ijk}, we can get:

$$\begin{aligned}g'_{ijk} &= \log(|\det(d\phi(i,j))| \cdot |\det(V_{ijk})| \cdot |\det(d\phi(i,j))|) \\ &= 2 \cdot \log(|\det(d\phi(i,j))|) + g_{ijk}\end{aligned}$$

$$\bar{g}'_{ij} = \frac{1}{C} \sum_{k=1}^{C} g'_{ijk} = 2 \cdot \log(|\det(d\phi(i,j))|) + \bar{g}_{ij}$$

Finally, computing U'_{ijk}:

$$U'_{ijk} = g'_{ijk} - \bar{g}'_{ij} = g_{ijk} - \bar{g}_{ij} = U_{ijk}$$

■

3. Experiments

We perform experiments on the CIFAR10 [11] dataset which contains low-resolution 32×32 color images of 10 different types of objects [11]. There is external environment noise in the background of each image. We evaluate the accuracy of the proposed DiffCNN model and compare its performance with (a) Affine Based (Spatial transformer) ST model [8], (b) CPAB ST model [7] and (c) RotDCF model [4]. Specifically, we train each model on 50000 clean images (without spatial transformation). No data augmentation for the training images was needed. Then, we evaluate the model on 10000 test images with three types of random spatial transformations. Each spatial transformation is across various scales of deviation $\sigma \in \{0.01, 0.05, 0.1, 0.5\}$ and centered at the identity matrix. The three types of spatial transformation matrices are as follows (transposed matrices are shown):

$$\underbrace{\begin{bmatrix} \mathcal{N}_{1,\sigma} & \mathcal{N}_{0,\sigma} \\ \mathcal{N}_{0,\sigma} & \mathcal{N}_{1,\sigma} \\ 0 & 0 \end{bmatrix}}_{\text{Affine Normal}}, \underbrace{\begin{bmatrix} \mathcal{N}_{1,\sigma} & \mathcal{N}_{0,\sigma} \\ \mathcal{N}_{0,\sigma} & \mathcal{N}_{1,\sigma} \\ \mathcal{N}_{0,\sigma} & \mathcal{N}_{0,\sigma} \end{bmatrix}}_{\text{Affine + Shift}}, \underbrace{\begin{bmatrix} \mathcal{N}_{1,\sigma} & \mathcal{N}_{0,\sigma} & \mathcal{N}_{0,\sigma} \\ \mathcal{N}_{0,\sigma} & \mathcal{N}_{1,\sigma} & \mathcal{N}_{0,\sigma} \\ \mathcal{N}_{0,\sigma} & \mathcal{N}_{0,\sigma} & 1 \end{bmatrix}}_{\text{Homography}}$$

where $\mathcal{N}_{\mu,\sigma}$ denotes the $\mathcal{N}(\mu, \sigma)$, the Gaussian distribution centered at μ and standard deviation of σ.

We use the CNN architectures with 7 hidden convolution layers [3] implemented by Chen and Smith as the base architecture for classification. This network is widely used and attains state-of-the-art classification performance. Then, our proposed DiffCNN structure, Affine Based ST model, CPAB ST model are separately added over the base classification architecture. For the RotDCF model, the convolution layers and batch normalization layers in the base classification architecture are replaced with the corresponding components from RotDCF model. These models are named for comparison as: DiffCNN + CNN, ST + CNN, CPAB ST + CNN and RotDCF + CNN.

During training process, the Adam optimizer [9] with a learning rate of 0.001 is trained for 100 epochs. Due to memory constraints, we selected training batch size and test batch size of 32 for the RotDCF + CNN model. For all other models, we used training batch size and test batch size of 100.

In order to assess the invariance behavior of our proposed DiffCNN model, we show classification accuracies with respect to the baseline models on the deformed images (Table 1). Some examples of these deformed images can be seen in Figure 3. It is clear from the table that DiffCNN consistently performs better than the standard CNN for several scales of deformed images. Further, there is a significant improvement, of up to 11%, for scales smaller than 0.1 relative to baselines. On the other hand, a small drop in performance of about 2% was observed on the undeformed (clean) images for the DiffCNN model. Interestingly, the table shows a small improvements even at extreme deformations such as at scale 0.5.

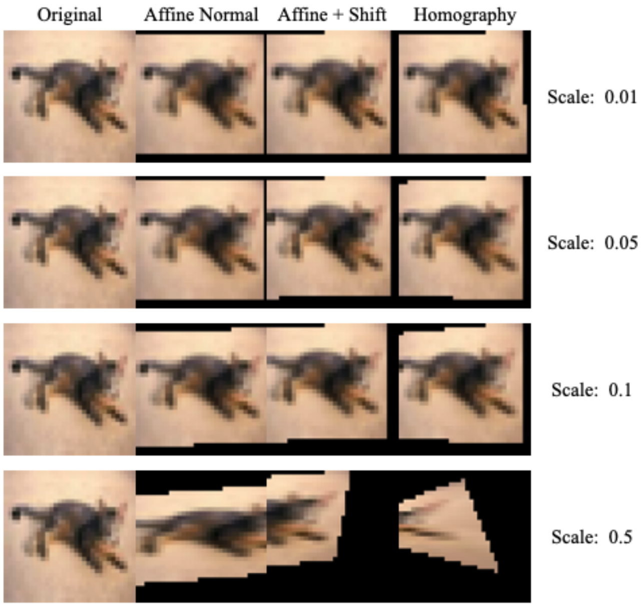

Figure 3. Illustration of the spatial transformations operated on an image of cat from the CIFAR10 dataset. The left most column is the original image without transformation. Except the left most column, each column represents different types of spatial transformation and each row represents different scales of the spatial transformation. Both the regular CNN and the DiffCNN classify the original image correctly as cat, while DiffCNN classifies 8 out of 12 deformed images correctly and the regular CNN only classifies 4 deformed images correctly.

4. Conclusions

We presented Diffeomorphism invariant Convolution Networks, DiffCNNs, that performs convolution on the Jacobian space of images. Our experiments on deformed images of CIFAR10 dataset suggest that the drop in performance in deformed images of various scales is much smaller for DiffCNN compared to a standard CNN. Further, based on preliminary experiments, DiffCNN appears competitive when compared to other state-of-the-art models for spatial transformation invariance from the literature. In summary, these results appear promising and provide a step forward for designing CNN models that operate well under different warpings of the image samples.

References

[1] Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research*, 21(90):1–61, 2020. 1

[2] Rudrasis Chakraborty, Monami Banerjee, and Baba C Vemuri. H-cnns: Convolutional neural networks for riemannian homogeneous spaces. *arXiv preprint arXiv:1805.05487*, 2:1, 2018. 1

[3] Xi Chen and Gus Smith. Pytorch playground. https://github.com/aaron-xichen/pytorch-playground. 3, 4

[4] Xiuyuan Cheng, Qiang Qiu, Robert Calderbank, and Guillermo Sapiro. Rotdcf: Decomposition of convolutional filters for rotation-equivariant deep networks. *arXiv preprint arXiv:1805.06846*, 2018. 1, 2, 3, 4

[5] Benjamin Chidester, Tianming Zhou, Minh N Do, and Jian Ma. Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, 35(14):i530–i537, 07 2019. 2

[6] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 1

[7] Nicki Skafte Detlefsen, Oren Freifeld, and Søren Hauberg. Deep diffeomorphic transformer networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4403–4412, 2018. 1, 2, 3, 4

[8] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015. 2, 3, 4

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[10] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755. PMLR, 2018. 1

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 4

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[14] Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. Medical image classification with convolutional neural network. In *2014 13th international conference on control automation robotics & vision (ICARCV)*, pages 844–848. IEEE, 2014. 1

[15] Robin Walters, Jinxi Li, and Rose Yu. Trajectory prediction using equivariant continuous convolution. *arXiv preprint arXiv:2010.11344*, 2020. 2

[16] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018. 2