Elastically-Augmented CNNs

Anonymous ICCV submission

Paper ID ****

Abstract

We focus on enhancing the performance of CNNs, especially against natural perturbations. Existing CNNs show outstanding performance on image classification, but training CNNs require large training datasets. We tackle this problem by incorporating elastic perturbations which approximate (local) view-point changes of the object in CNNs. We present elastically-augmented convolutions (EA-Conv) by parameterizing filters as a combination of fixed elastically-perturbed bases functions and trainable weights. We show an improvement in the general robustness of CNNs on the CIFAR-10 and STL-10 datasets, while even improving the performance on clean images without any data augmentation.

1. Introduction

Convolutional Neural Networks (CNNs) achieve state-of-the-art performance on image classification tasks, however learning powerful features using CNNs require large amounts of training dataset [11, 6, 8]. One way to solve this problem is to pretrain on a similar dataset to the test set [15]. However, in the real-world, large datasets similar to the target domain are not always available for pretraining. Yet another option is to incorporate prior geometric knowledge in the network [8, 12].

A straight forward method to incorporate prior knowledge in the training framework is through data augmentation. However, here we assume there are no more images or their transformed versions are available. We begin with the argument that in practice frequent and important deformations occur in the image when the camera changes its viewing angle, when the lighting conditions are not suitable which leads to the noisy image, and when the object is partially hidden. Hence, it is important to test the robustness of networks against such perturbations before deploying them in the real world. For all such perturbations, elastic transforms will provide a better variation of input features to the next layer.

Global elastic transformations correspond to view point changes, and local elastic transformations cover out-of-plane rotations of the object. Typical examples are in medical images, when the shape and volume of organs may vary [16, 1]. Similar behavior is observed when the object is still and the scene is dynamic as in ocean waves, or when the camera moves as in video segmentation or tracking. We pronounce that local elastic transforms provide a good approximation of many practical local variations in the image space one wants to be invariant under.

Our contributions are: (i). We propose the theory for elastically-augmented convolutional neural networks. (ii). We introduce *Elastically-Augmented Convolutions* to integrate unseen viewpoints in the CNNs for enhancing their general robustness. (iii). We demonstrate that by incorporating local elastic variations in the convolutions of the network, we enhance the performance on clean images. (iv). We demonstrate *specific* robustness for elastic transforms and, remarkably, *general* robustness for Gaussian and occlusion perturbations unseen during training.

2. Related Work

Built-in Image Transformations. Initially, geometric transformations were modeled in the neural networks by small units that locally transformed their inputs for modeling geometric changes, i.e. capsules [7]. Later [9] introduced a transformer module in the network to wrap feature maps by global transformations. However, learning the parameters of the transformations introduced by [9] is known to be difficult and computationally expensive. In similar spirit, [4, 3] focused on integrating spatial deformations in CNNs. Both methods require large datasets for learning, while our aim is to learn from small datasets and generalize the performance to include perturbations on images never seen before.

Robustness to Natural Perturbations To improve the robustness against natural perturbations, [14] introduced a noise generator that learns uncorrelated noise distributions. Training on these noisy images enhanced the performance against natural perturbations. [5] trained on images with adversarial as well as natural perturbations, while achieving good generalization for unseen perturbations. [13, 17]

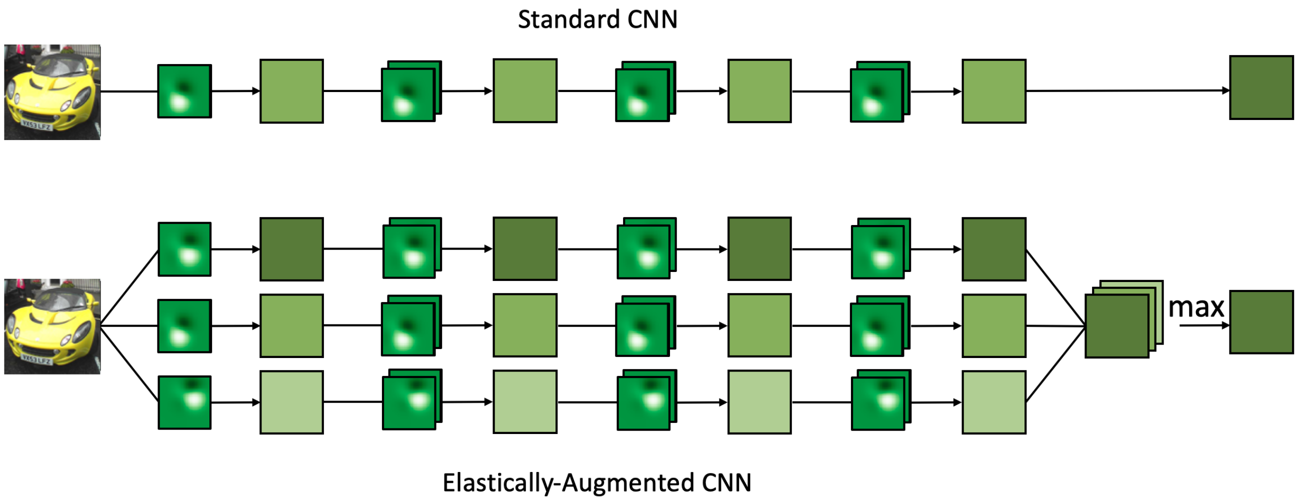

Figure 1. An illustration of a standard CNN with 4 convolutional layers and its elastically-augmented variant.

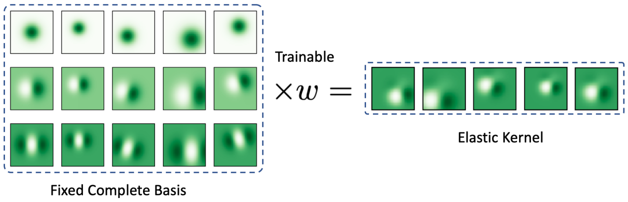

Figure 2. An illustration of how a set of elastic kernels is represented as a trainable linear combination of elastically-augmented fixed basis functions.

argued that it is impossible to capture all possible natural perturbations mathematically. Therefore, they used generative models to generate images with perturbations to train the network. Instead of training with perturbed inputs, in this work we integrate predefined perturbations into the network to enhance robustness.

3. Method

3.1. Image Transformations

Consider an image f. It can be reshaped as a vector $\mathbf{f}$. A wide range of image transformations can be parametrized by a linear operator: scaling, in-plane rotations, shearing. Other transformations, such as out-of-plane rotations, can not be parametrized in an image agnostic way. However, for small deviation from the original image Taylor expansions can be used, which gives a linear approximation for many image transformations of practical use. Indeed,

$$\begin{aligned} T[f](\epsilon) &\approx T[f](0) + \epsilon\left(\frac{\partial T[f]}{\partial \epsilon}\right)\Bigg|_{\epsilon=0} = \mathbf{f} + \epsilon\mathbf{L}_T \times \mathbf{f} \\ &= (\mathbf{I} + \epsilon\mathbf{L}_T) \times \mathbf{f} = \mathbf{T} \times \mathbf{f} \end{aligned} \tag{1}$$

where T is a transformation, ϵ is the parameter of the transformation and $\mathbf{T}$ is a linear approximation of T for small values of the parameter. For scaling the parameter is the logarithm of the scaling factor, for rotations it is the angle, and so on. $\mathbf{L}_T$ is a matrix representation of an infinitesimal generator of T.

An image f can also be viewed as a real-value function of its coordinates $f : x \to f(x)$. We focus here on transformations which can be represented by a smooth field of displacements τ in the space of coordinates. Equation 1 can then be rewritten as follows:

$$T[f(x)](\epsilon) \approx f(x + \epsilon\tau(x)) \tag{2}$$

We will refer to such transformations as elastic transformations. We will consider them as a linear approximation of a wide range of complex (camera) transformations.

3.2. Elastically-Augmented Convolutions

Let us consider a convolutional layer Φ parameterized by a filter κ and an input image f. The output is:

$$\Phi(f, \kappa) = f \star \kappa = \mathbf{K} \times \mathbf{f} \tag{3}$$

where $\mathbf{K}$ is a matrix representation of the filter. While, when data augmentation is used, a transformed version of the image can be fed as an input.

$$\begin{aligned} \Phi(T[f], \kappa) &= T[f] \star \kappa = \mathbf{K} \times (\mathbf{T} \times \mathbf{f}) \\ &= (\mathbf{K} \times \mathbf{T}) \times \mathbf{f} = \Phi(f, T'[\kappa]) \end{aligned} \tag{4}$$

In general, when $\mathbf{KT}$ is a matrix representation of some kernel, the result can be achieved by transforming the kernel instead of transforming the input.

To incorporate the data augmentation into the convolutional layers of the network, we propose *elastically-*

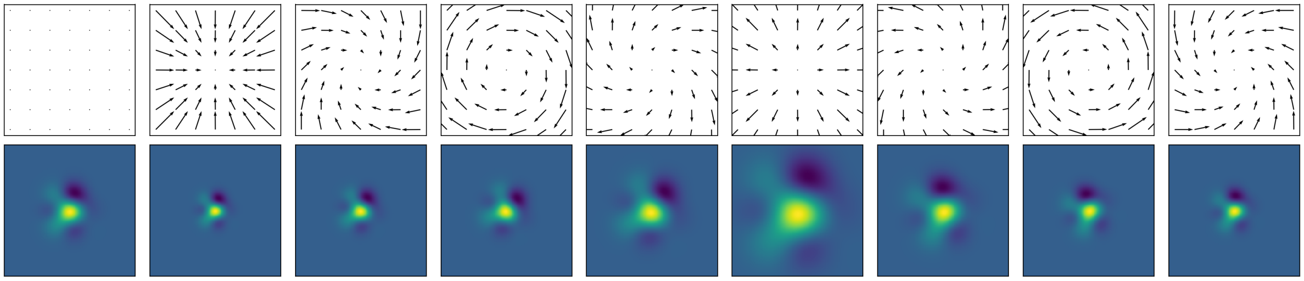

Figure 3. Top: vector fields of smooth displacements for the proposed set of rotation-scaling transformations. Bottom: the original filter and its versions transformed after applying the corresponding displacements.

Model	Clean	Perturbed-1	Perturbed-2
Standard Network	0.00	-5.89	-10.53
EAConv Network	0.00	$+3.68$	$+6.36$
Data Augmentation	-1.57	$+4.07$	$+8.02$

Table 1. Comparing a standard network and EAConv Network for CIFAR-10 clean and perturbed inputs. Negative values show the drop in the performance, and positive values show recovery in the drop.

augmented convolutions, shortly EAConv, as follows:

$$\text{EAConv} = \max \begin{bmatrix} \beta_0 \Phi(f, \kappa) \\ \beta_1 \Phi(f, T_1[\kappa]) \\ \vdots \\ \beta_n \Phi(f, T_n[\kappa]) \end{bmatrix} \tag{5}$$

where β_i are trainable coefficients. We initialize them such that $\beta_0 = 1$ and the rest are zeros. The maximum is calculated per pixel among different transformations of the kernel. At the beginning of training, the operation is thus identical to the original convolution with the same filter. If it is required during training, the other coefficients will activate the corresponding transformations.

3.3. Transformations of a Complete Basis

In order to apply elastic transformations to filters, we parametrize each filter as a linear combination of basis functions:

$$\kappa = \sum_i w_i \psi_i \tag{6}$$

where ψ_i are functions of a complete fixed basis and w_i are trainable parameters. The approach is illustrated in Figure 2. We follow [8] and choose a basis of 2-dimensional Gaussian derivatives.

The transformations when applied to the basis form a transformed basis. Thus, for every transformation from the set, there is a corresponding transformed basis. Weights w_i are shared among all bases.

Let us assume that the center of a filter is a point with coordinates $(0, 0)$. For every function from the basis, we first generate a grid of coordinates (x, y). Then we evaluate the value of the function in the coordinates when projected on the pixel grid. In order to transform the functions, we add a small displacement to the coordinates, which leaves the center untransformed. We propose a set of transformations which we call rotations-scaling displacements. See Figure 3.

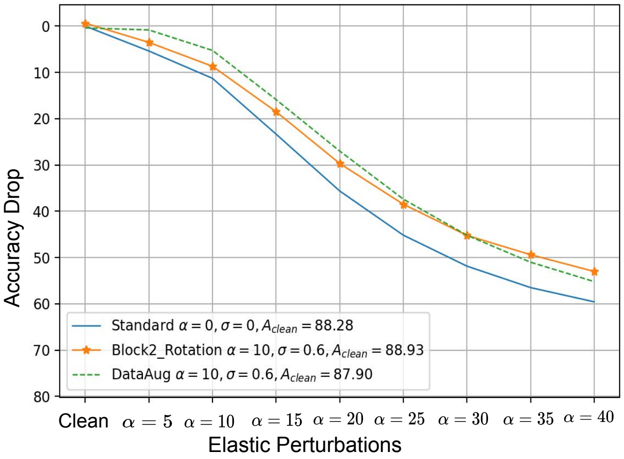

Figure 4. Evaluating the performance of elastically-augmented convolutions on elastic perturbations at different levels of severity on x-axis.

3.4. Elastically-Augmented Residual blocks

In order to transform residual networks, we propose a straightforward generalization of the proposed convolution. The standard residual block can be formulated as follows:

$$\text{ResBlock} = f + G(f, \kappa_1, \kappa_2, \dots) \tag{7}$$

The according augmented block is formulated as follows:

$$\text{EAResBlock} = f + \max \begin{bmatrix} \beta_0 G(f, \kappa_1, \kappa_2, \dots) \\ \beta_1 G(f, T_1[\kappa_1], T_1[\kappa_2], \dots) \\ \vdots \\ \beta_n G(f, T_n[\kappa_1], T_n[\kappa_2], \dots) \end{bmatrix} \tag{8}$$

Elastic kernels augmented in the network architecture are shown in the Figure.1.

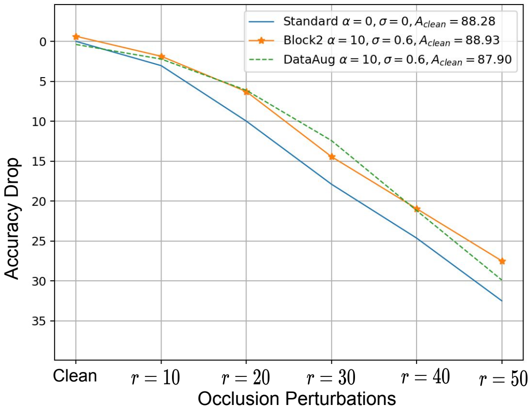

Figure 5. Evaluating EAConv on unseen occlusion perturbations.

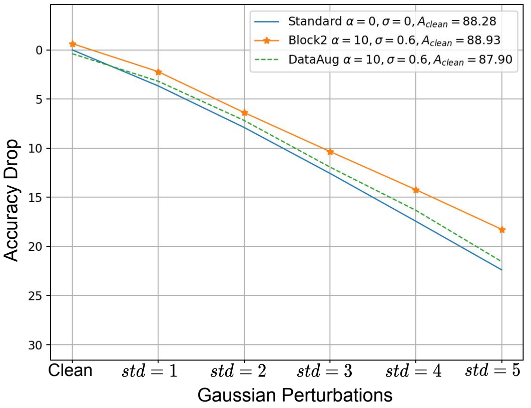

Figure 6. Evaluating EAConv on unseen Gaussian perturbations.

4. Experiments and Results

4.1. Standard Network

We began by training and testing standard networks for CIFAR-10 [10] and STL-10 datasets [2] respectively on clean images. For CIFAR-10, we trained a Resnet-50 network and achieved 91.11 and for STL-10, we trained a Wide-Resnet-16 and gained 88.28 on clean test set.

4.2. Elastically-Augmented Convolutional Network

Elastic CIFAR-10. We experimented by augmenting different layers of the network with EAConv and selected the combination which gave the best performance on clean samples, i.e. the first convolutional layer: 91.11.

Table.1 contrasts the performance of a standard network with our elastically augmented network on CIFAR-10 clean images as well as on elastic perturbed images causing a drop of 5.89 and 10.53. Results show that our model without any drop in the performance on clean images leads to a gain in the performance on elastic perturbed images. On the other hand, although data augmentation helps against perturbed inputs, however it leads to a drop on clean images.

Elastic STL-10. For STL-10, we also experimented by augmenting different layers of a Wide-Resnet-16 with EAConv. We selected the augmented combination of layers which gave the best performance on clean images, i.e. layers till Block2: 88.93.

Evaluating on Seen Perturbations. In Figure. 4 x-axis, we have a clean test set and eight elastically perturbed test sets with varying severity levels. While on the y-axis, we have the drop in the accuracy with the clean test on a standard network with the drop zero and increasing drop with the increase in the severity levels. The blue solid line shows the drop on a standard network for elastically perturbed samples.

Results show that with the increase in the perturbations the accuracy drops (blue solid line), however, our EAConv network especially rotation scaling transforms recover the drop for all the severity levels while enhancing the performance on clean test set (orange line with star symbols). Although data augmentation helps against elastic perturbations, but it leads to a drop in the performance on clean images (green dotted line).

Evaluating on Unseen Perturbations. Figure. 5 shows the performance of networks tested on a clean and five different occluded test sets with varying sizes of occlusion from radius $r = 10$ to 50. We observe that the classification accuracy drops with the increase in the size of occlusion on a standard network (blue solid line). However, our EAConv network (Orange solid line with star symbols) shows recovery in the drop, hence generalizing to unseen occlusion perturbations. Data augmentation with elastic perturbations shows robustness for some occluded test sets, but for others the improvement is less than our method.

Figure. 6 shows the effectiveness of our method on unseen Gaussian perturbations. On the x-axis, we have a clean test set and five different test sets with the Gaussian perturbations induced with the varying standard deviation $std = 1$ to 5. The plots show that with the increase in the severity of Gaussian noise, the accuracy drops for a standard network (blue solid line). However, when we test our EAConv on these perturbations, it helps to recover the drop. In contrast, data augmentation with elastic deformations show very minimal improvement in the performance.

5. Conclusion

A method to integrate unseen view points in CNNs is introduced for enhancing the robustness against local variations in the image space. We demonstrated the effectiveness of our method by improving the performance on perturbed inputs without the loss of generality on clean inputs. We also showed the general robustness of our EAConv network by testing on unseen occlusion and Gaussian perturbations. Our results showed that elastically augmented convolutions enhance the robustness against unseen viewpoint variations while keeping the number of training parameters in the network and the number of training images the same.

References

[1] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020. 1

[2] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 4

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 1

[4] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. 1

[5] Sadaf Gulshad and Arnold Smeulders. Natural perturbed training for general robustness of neural network classifiers. *arXiv preprint arXiv:2103.11372*, 2021. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[7] Geoffrey F Hinton. A parallel computation that assigns canonical object-based frames of reference. In *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2*, pages 683–685, 1981. 1

[8] Jorn-Henrik Jacobsen, Jan Van Gemert, Zhongyu Lou, and Arnold WM Smeulders. Structured receptive fields in cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2610–2619, 2016. 1, 3

[9] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015. 1

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

[11] 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

[12] Matthias Rath and Alexandru Paul Condurache. Boosting deep neural networks with geometrical prior knowledge: A survey. *arXiv preprint arXiv:2006.16867*, 2020. 1

[13] Alexander Robey, Hamed Hassani, and George J Pappas. Model-based robust deep learning. *arXiv preprint arXiv:2005.10247*, 2020. 1

[14] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision*, pages 53–69. Springer, 2020. 1

[15] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. 1

[16] Xiu Ying Wang, David Dagan Feng, and Jesse Jin. Elastic medical image registration based on image intensity. In *Proceedings of the Pan-Sydney area workshop on Visual information processing-Volume 11*, pages 139–142. Citeseer, 2001. 1

[17] Eric Wong and J Zico Kolter. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv:2007.08450*, 2020. 1