Review: Spatial Transformer Networks

Jiho Bak
School of Electrical and Electronic Engineering, Yonsei University

Abstract

The spatial invariance is a limitation of convolutional neural network(CNN). For CNN, this spatial invariance can be only realized by deep-hierarchy of max-pooling and convolutions. Also through out researches (e.g. [2], [3]), they found out that feature maps in CNN are not actually spatially invariant. The spatial transformer module is proposed for this limitation. This module is differentiable(which enables backpropagation) and conditional on the input feature map. Also it can be inserted in any part of the network and trained in end-to-end framework. This spatial transformer module can learn invariance to translation, rotation, scale and other warping, which can be seen as a generalization of differentiable attention to any spatial transformation. These properties leads the spatial transformer networks to the state-of-the-art performance for several tasks.

1 Spatial Transformer

As introduced, spatial transformer is a module which actually transforms the input feature map, which conventional cNNs are not capable of. The spatial transformer is realized in three procedures. The localization network, grid generator and sampler. The process is visualized in Fig 1.

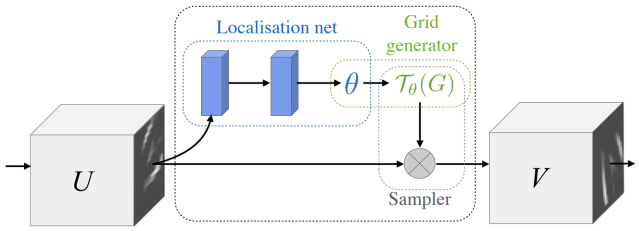

Figure 1: The architecture of the spatial transformer module.

1.1 Localization Network

The localization network takes the input image or feature map and outputs the parameters for transformation. This network can be any form, in this case, neural network.

$$\theta = f_{loc}(U), \quad \text{where } U \in R^{H\times W\times C}$$

This network enables the input conditional transformation.

1.2 Parameterized Sampling Grid

With the parameter θ, derived from localization network, the sampling grid is generated. This grid generator can change the heights and width from input while remaining the number of channel.
The transformation T_θ, maps the original source coordinate (x_i^t, y_i^t) to target coordinate (x_i^s, y_i^s) where they are normalized to $-1 \le x_i^t, y_i^t, x_i^s, y_i^s \le 1$.
The below (1) is a 2D affine transformation, and the comparison between identity, affine transformation is in Fig2. This generalized transformation allows cropping, rotation, scale, translation and skew. The transformation can be controlled by regularizing θ. each affine, projective and thin plate spline transformations are used for experiment.

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{1}$$

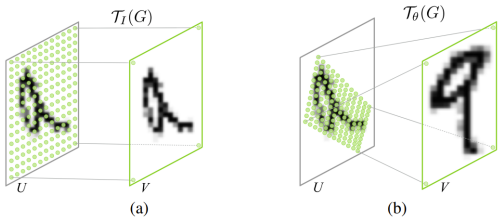

Figure 2: The sampling grid examples with (a) identity transformation, and (b) affine transformation.

1.3 Differentiable Image Sampling

To actually sample the value from the input image, not only with coordinates, the sampling function should be differentiable. So, the bilinear sampling kernel is used as (2). Then, the partial derivative for each U, x, y is (4), (5), (6). This is sub-differentiable sampling. While backpropagation, the gradient of U flows through the convolution layers, and the gradient of x, y flow through the localization network by $\frac{\partial x}{\partial \theta}$.

$$V_i^C = \sum_n^H \sum_m^W U_{nm}^C \max(0, 1-|x_i^s - m|) \max(0, 1-|y_i^s - n|) \tag{2}$$

$$\frac{\partial V_i^C}{\partial U_{nm}^C} = \sum_n^H \sum_m^W \max(0, 1-|x_i^s - m|) \max(0, 1-|y_i^s - n|) \tag{3}$$

$$\frac{\partial V_i^C}{\partial x_i^s} = \sum_n^H \sum_m^W U_{nm}^C \max(0, 1-|y_i^s - n|) \begin{cases} 0 & \text{if } |m - x_i^s| \ge 1 \\ 1 & \text{if } m \ge x_i^s \\ -1 & \text{if } m \le x_i^s \end{cases} \tag{4}$$

$$\frac{\partial V_i^C}{\partial y_i^s} = \sum_n^H \sum_m^W U_{nm}^C \max(0, 1-|x_i^s - n|) \begin{cases} 0 & \text{if } |m - y_i^s| \ge 1 \\ 1 & \text{if } m \ge y_i^s \\ -1 & \text{if } m \le y_i^s \end{cases} \tag{5}$$

1.4 Spatial Transformer Networks

The spatial transformer module inside the network, actively transforms the feature maps. The author claims that the module not only helps the optimizing loss function, but also is computationally fast which doesn't degrades the training speed. In addition, the transformation information could be explicitly encoded to the network, by feeding θ to other layers. And the author says that small sampling kernel should be avoided since it can cause aliasing.
Finally, author proposes the multiple spatial transformers which increases abstract and informative representation. Also the parallel transformer can help for tasks with multiple objects/interests. However, this architecture shows pure limitation that the number of the transformer determines the number of objects that model can deal with.

2 Experiment

In paper, three different experiments are introduced. First experiment is showing the active transformation ability of the module, which improves classification performance, with distored MNIST dataset. The second is for the performance in real-world data, which showed the state-of-the-art result with Street View House Number [4] dataset. The last is the investigation of multiple parallel spatial transformers for fine-grained classification. This experiment was done with CUB [1] dataset and also showed the state-of-the-art performance.

2.1 Distorted MNIST: How spatial transformation actually works

To explore how the module actually transforms the image, various distortion was applied to the data. Rotation(R), rotationtranslationscale(RTS), projective transformation(P), and elastic warping(E), which is the most destructive. For comparison, fully-connected neural network and convolutional neural network were set as baseline. And in case of the spatial transformer network, affine(Aff), projective(Proj) and 16-point thin plate spline(TPS) transformation were respectively tested. The result is in Fig 3.
ST-CNN performed best, and for transformation, TPS performed the best. The performance of ST-FCN shows that using a spatial transformer can be alternative method for spatial invariance. The all ST models transformed the image into upright posed digit which was the mean pose of train dataset. Few examples are shown in Fig 3. As a result, the transformation that the module has done actually increase spatial invariance and the classification performance.

Model		MNIST Distortion R	RTS	P	E
FCN		2.1	5.2	3.1	3.2
CNN		1.2	0.8	1.5	1.4
ST-FCN	Aff	1.2	0.8	1.5	2.7
	Proj	1.3	0.9	1.4	2.6
	TPS	1.1	0.8	1.4	2.4
ST-CNN	Aff	0.7	0.5	0.8	1.2
	Proj	0.8	0.6	0.8	1.3
	TPS	0.7	0.5	0.8	1.1

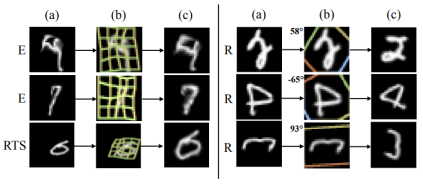

Figure 3: The experiment result with MNIST dataset.

2.2 Street View House Numbers: Performance with real-world dataset

The task with this dataset is predicting the house numbers which is 1-5 digits. Every networks used five independent softmax classifiers for particular prediction in sequence. For the model, ST-CNN network with single and multiple modules were both designed for experiment. ST-CNN Single used 4-layer CNN for localization network, and ST-CNN Multi used the modules consist of 2 fully connected layers(with 32 units each). Affine transformation was used for both.
ST-CNN models both performed better than other sota models, and Multi module model performed best. Comparing with other sota models, ST models was better with larger scale(128×128) inputs. Also ST models required only a single forward pass, unlike with other ensemble models.
For ST-CNN Multi model, the first module act just like in the Single version. But the subsequent modules was predicting the transformation in the deeper feature maps. This deeper spatial transformation is dealing the richer features.(Fig 4, (b))

Model		Size 64px	128px
Maxout CNN [13]		4.0	-
CNN (ours)		4.0	5.6
DRAM* [1]		3.9	4.5
ST-CNN	Single	3.7	**3.9**
	Multi	**3.6**	**3.9**

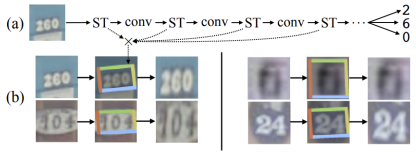

Figure 4: The experiment result with Street Veiw House Numbers dataset.

2.3 Fine-Grained Classification: About parallel transformer

The last experiment was for testing the parallel multiple transformers. The fine-grained bird classification was designed with Inception architecture as a baseline CNN model. ST-CNN was designed with 2 or 4 parallel spatial transformers, The resulting part are concatenated before softmax layer. The whole network was trained in end-to-end and compared to sota models.
All ST model out-performed the sota models about $1 \sim 2$ % (Fig 5). Interesting observation was that each spatial transformer tend to detect certain part of the bird e.g. head, body. As a result, without any other supervision, parallel modules learned pose-normalized representation as a part detector. Which led to the state-of-the-art performance.

Model	
Cimpoi '15 [5]	66.7
Zhang '14 [40]	74.9
Branson '14 [3]	75.7
Lin '15 [23]	80.9
Simon '15 [30]	81.0
CNN (ours) 224px	82.3
2×ST-CNN 224px	83.1
2×ST-CNN 448px	83.9
4×ST-CNN 448px	**84.1**

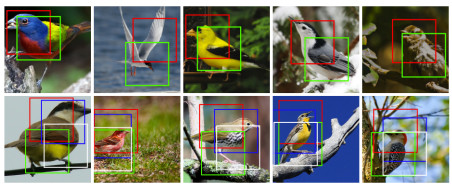

Figure 5: The experiment result with CUB dataset.

3 Conclusion

The spatial transformer network, which enabled the spatial invariance in data-driven manner, out-performed the sota models in various tasks. Also, since the training process is end-to-end and relatively fast, we can confirmed the effectiveness of this concept. Furthermore, there are more room for utilization, e.g. using the regressed transformation parameters, applying in recurrent models, 3D transformation(in appendix).

Reference

[1] P. Welinder P. Perona C. Wah, S. Branson and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[2] T. S. Cohen and M. Welling. Transformation properties of learned visual representations. *ICLR*.

[3] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. *ICLR*.

[4] A. Coates A. Bissacco B. Wu Y. Netzer, T. Wang and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. *NIPS DLW*.