Spatial Transformer Networks

Max Jaderberg
Andrew Zisserman
Karen Simonyan
Koray Kavukcuoglu

Okay ARIK
Contents

• Introduction to Spatial Transformers
• Related Works
• Spatial Transformers Structure
• Spatial Transformer Networks
• Experiments
• Conclusion
Introduction

• CNNs have lack of ability to be spatial invariance in a computationally and parameter efficient manner.
• Max-pooling layers in CNNs satisfy this property where the receptive fields are fixed and local.
• Spatial transformer module is a dynamic mechanism that can actively spatially transform an image or a feature map.
Introduction

• Transformation is performed on the entire feature map (non-locally) and can include scaling, cropping, rotations, as well as non-rigid deformations.

• This allows networks to not only select regions that are most relevant (attention), but also to transform those regions.
Introduction

• Spatial transformers can be trained with standard back-propagation, allowing for end-to-end training of the models they are injected in.

• Spatial transformers can be incorporated into CNNs to benefit multifarious tasks:
  ▪ *image classification*
  ▪ *co-localisation*
  ▪ *spatial attention*
Related Works

• Hinton (1981) looked at assigning canonical frames of reference to object parts, where 2D affine transformations were modeled to create a generative model composed of transformed parts.
Related Works

- Lenc and Vedaldi studied **invariance and equivariance** of CNN representations to input **image transformations** by estimating the linear relationships.

- Gregor et al. use a **differentiable attention mechanism** by utilising Gaussian kernels in a generative model. **This paper generalizes differentiable attention to any spatial transformation.**
Spatial Transformer

• **Spatial transformer** is a differentiable module which applies a spatial transformation to a feature map and produces a single output feature map.

• For multi-channel inputs, the same warping is applied to each channel.
Spatial Transformer

• The spatial transformer mechanism is split into three parts:
Spatial Transformer

- **Localisation network** takes the input feature map, and through a number of hidden layers outputs parameters of spatial transformation.
Spatial Transformer

- **Grid generator** creates a **sampling grid** by using predicted transformation parameters.
Spatial Transformer

- **Sampler** takes feature map and the sampling grid as inputs, and produces the output map sampled from the input at the grid points.
Spatial Transformer

• **Localisation network** takes the input feature map and outputs parameter $\theta$ for the transformation.

• Size of $\theta$ can vary depending on the transformation type that is parameterised.
Spatial Transformer

• **Grid Generator:** *Identity transformation*

Output pixels are defined to lie on a regular grid.
Spatial Transformer

• **Grid Generator:** *Affine Transform*

Output pixels are defined to lie on a regular grid.

**Sampling Grid**
Spatial Transformer

- **Grid Generator:** *Affine Transform*

\[
\begin{pmatrix}
  x_i^s \\
  y_i^s
\end{pmatrix}
= \mathcal{T}_\theta(G_i) = A_\theta
\begin{pmatrix}
  x_i^t \\
  y_i^t \\
  1
\end{pmatrix}
= \begin{bmatrix}
  \theta_{11} & \theta_{12} & \theta_{13} \\
  \theta_{21} & \theta_{22} & \theta_{23}
\end{bmatrix}
\begin{pmatrix}
  x_i^t \\
  y_i^t \\
  1
\end{pmatrix}
\]
Spatial Transformer

• Differentiable Image Sampling

Any sampling kernel

\[
V_i^c = \sum_{n}^{H} \sum_{m}^{W} U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y)
\]

- target value
- source value
- sampling grid coordinate (not integer necessarily)
Spatial Transformer

• Differentiable Image Sampling

Integer sampling

\[ V_i^c = \sum_{n}^{H} \sum_{m}^{W} U_{nm}^c \delta([x^s_i + 0.5] - m)\delta([y^s_i + 0.5] - n) \]

- target value
- source value
- sampling grid coordinate (not integer necessarily)
Spatial Transformer

- Differentiable Image Sampling

Bilinear sampling

\[ V_i^c = \sum_{n}^{H} \sum_{m}^{W} U_{nm}^c \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|) \]

- Target value
- Source value
- Sampling grid coordinate (not integer necessarily)
Spatial Transformer

- Differentiable Image Sampling

To allow backpropagation of the loss through this sampling mechanism, gradients with respect to U and G can be defined as:

\[
\frac{\partial V^c_i}{\partial U^c_{nm}} = \sum_n \sum_m \max(0, 1 - |x^s_i - m|) \max(0, 1 - |y^s_i - n|)
\]

\[
\frac{\partial V^c_i}{\partial x^s_i} = \sum_n \sum_m U^c_{nm} \max(0, 1 - |y^s_i - n|) \begin{cases} 
0 & \text{if } |m - x^s_i| \geq 1 \\
1 & \text{if } m \geq x^s_i \\
-1 & \text{if } m < x^s_i 
\end{cases}
\]
Spatial Transformer Networks

• Placing spatial transformers within a CNN allows the network to learn how to actively transform the feature maps to help minimise the overall cost function of the network during training.

• The knowledge of how to transform each training sample is compressed and cached in the weights of the localisation network.
Spatial Transformer Networks

• For some tasks, it may also be useful to feed the output of the localisation network $\theta$, forward to the rest of the network, as it explicitly encodes the transformation, and hence the **pose of a region or object**.

• It is possible to use spatial transformers to downsample or oversample a feature map.
Spatial Transformer Networks

• It is possible to have **multiple spatial transformers** in a CNN.

• Multiple spatial transformers **in parallel** can be useful if there are **multiple objects or parts** of interest in a feature map that should be focussed on individually.
Experiments

• Distorted versions of the MNIST handwriting dataset for classification
• A challenging real-world dataset, Street View House Numbers for number recognition
• CUB-200-2011 birds dataset for fine-grained classification by using multiple parallel spatial transformers
Experiments

• MNIST data that has been distorted in various ways: rotation (R), rotation, scale and translation (RTS), projective transformation (P), and elastic warping (E).

• Baseline fully-connected (FCN) and convolutional (CNN) neural networks are trained, as well as networks with spatial transformers acting on the input before the classification network (ST-FCN and ST-CNN).
Experiments

- The spatial transformer networks all use different transformation functions: an affine (Aff), projective (Proj), and a 16-point thin plate spline transformations (TPS)
Experiments

- **Affine Transform** (error %)

![Bar Chart](chart.png)

Okay ARIK
Experiments

• **Projective Transform** (error %)

![Bar chart showing projective transform error percentages for different categories: R, RTS, P, and E. The chart compares CNN and ST-CNN models.](chart.png)
Experiments

- **Thin Plate Spline** (error %)

![Bar chart showing error percentages for Thin Plate Spline]
Experiments

- **Street View House Numbers (SVHN)**
- This dataset contains around 200k real world images of house numbers, with the task to recognise the sequence of numbers in each image
Experiments

- Data is preprocessed by taking $64 \times 64$ crops and more loosely $128 \times 128$ crops around each digit sequence.
Experiments

- Comparative results (error %)

<table>
<thead>
<tr>
<th></th>
<th>Maxout CNN</th>
<th>Ours</th>
<th>DRAM</th>
<th>ST-CNN Single</th>
<th>ST-CNN Multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>4</td>
<td>4</td>
<td>3.9</td>
<td>3.7</td>
<td>3.6</td>
</tr>
<tr>
<td>128</td>
<td></td>
<td></td>
<td>5.6</td>
<td>3.9</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Okay ARIK
Experiments

• Fine-Grained Classification
• CUB-200-2011 birds dataset contains 6k training images and 5.8k test images, covering 200 species of birds.
• The birds appear at a range of scales and orientations, are not tightly cropped.
• Only image class labels are used for training.
Experiments

• Baseline CNN model is an Inception architecture with batch normalisation pretrained on ImageNet and fine-tuned on CUB.
• It achieved the state-of-the-art accuracy of 82.3% (previous best result is 81.0%).
• Then, spatial transformer network, ST-CNN, which contains 2 or 4 parallel spatial transformers are trained.
Experiments

• The transformation predicted by 2×ST-CNN (top row) and 4×ST-CNN (bottom row)
Experiments

• One of the transformers learns to detect heads, while the other detects the body.
Experiments

- The accuracy on CUB (%)
Conclusion

• We introduced a new self-contained module for neural networks.
• We see gains in accuracy using spatial transformers resulting in state-of-the-art performance.
• Regressed transformation parameters from the spatial transformer are available as an output and could be used for subsequent tasks.