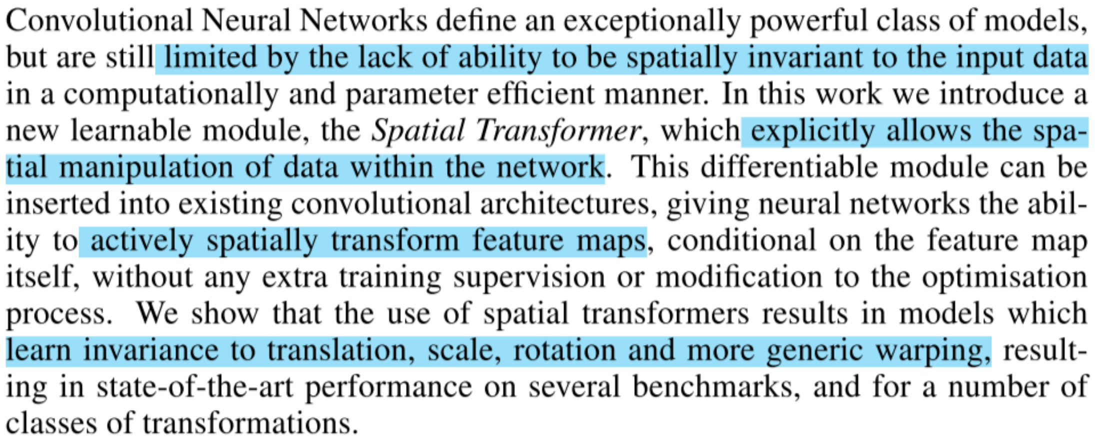
STN 阅读总结

15331416 赵寒旭

Abstract



Convolutional Neural Networks define an exceptionally powerful class of models, but are still limited by the lack of ability to be spatially invariant to the input data in a computationally and parameter efficient manner. In this work we introduce a new learnable module, the *Spatial Transformer*, which explicitly allows the spatial manipulation of data within the network. This differentiable module can be inserted into existing convolutional architectures, giving neural networks the ability to actively spatially transform feature maps, conditional on the feature map itself, without any extra training supervision or modification to the optimisation process. We show that the use of spatial transformers results in models which learn invariance to translation, scale, rotation and more generic warping, resulting in state-of-the-art performance on several benchmarks, and for a number of classes of transformations.

卷积神经网络定义了一个强大的分类模型，但是缺乏应对输入数据空间变换的能力。我们介绍了一种新的可习得的模型——空间变换，它明确允许了网络中数据的空间操作。这一可微的模型可以被插入目前的卷积结构，使神经网络能够对 feature map 进行基于其自身的空间变换而不需要对 optimisation process 进行任何额外的监督或修改。空间变换模型能够学习到平移，放缩，旋转和更多通用变换的不变性，提高网络的性能。

Q: spatial invariance 具体含义是否指即使一个物体在空间上发生了变换，但是物体的属性仍然不变？比如原来的图像是数字 1，即使经过平移旋转缩放等一系列空间变换，仍然保持为数字 1 并应该被网络识别。

试解：空间不变性对应着图像处理的经典手段：平移、缩放和旋转，他们同属于空间变换，并可以通过坐标矩阵的仿射变换来实现。

仿射变换矩阵：

1. 平移

$$\begin{bmatrix} 1 & 0 & \theta_{13} \\ 0 & 1 & \theta_{23} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + \theta_{13} \\ y + \theta_{23} \end{bmatrix}$$

2. 缩放

$$\begin{bmatrix} \theta_{11} & 0 & 0 \\ 0 & \theta_{23} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_{11} x \\ \theta_{22} y \end{bmatrix}$$

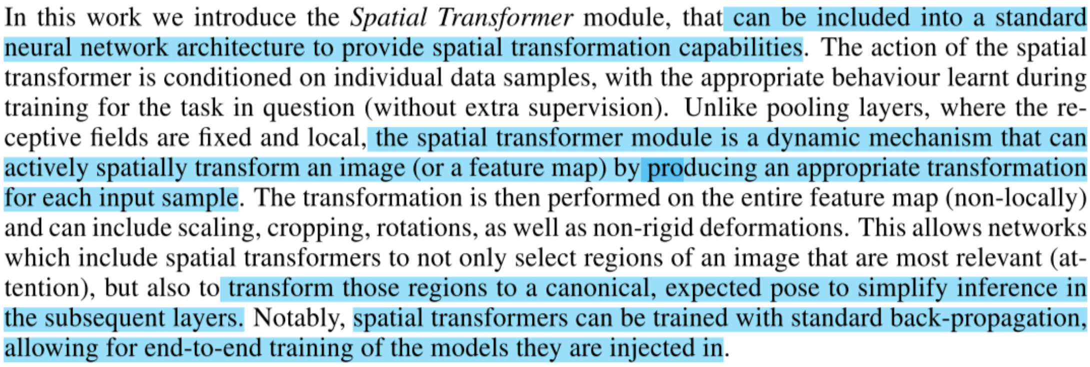
3. 旋转

绕原点顺时针旋转α度，坐标仿射矩阵：

$$\begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha)\,x + \sin(\alpha)\,y \\ -\sin(\alpha)\,x + \cos(\alpha)\,y \end{bmatrix}$$

需要做 Normalization 把坐标调整到$[-1,1]$使绕图像中心旋转。

1. Introduction



In this work we introduce the *Spatial Transformer* module, that can be included into a standard neural network architecture to provide spatial transformation capabilities. The action of the spatial transformer is conditioned on individual data samples, with the appropriate behaviour learnt during training for the task in question (without extra supervision). Unlike pooling layers, where the receptive fields are fixed and local, the spatial transformer module is a dynamic mechanism that can actively spatially transform an image (or a feature map) by producing an appropriate transformation for each input sample. The transformation is then performed on the entire feature map (non-locally) and can include scaling, cropping, rotations, as well as non-rigid deformations. This allows networks which include spatial transformers to not only select regions of an image that are most relevant (attention), but also to transform those regions to a canonical, expected pose to simplify inference in the subsequent layers. Notably, spatial transformers can be trained with standard back-propagation, allowing for end-to-end training of the models they are injected in.

空间变换器模块可以被包含在标准神经网络结构中以提供空间变换能力。

空间变换器模块是一种动态机制，可以通过对每个输入样本产生合适的变换对一个图像（或者一个 feature map）进行主动的空间变换。

包含空间变换的网络将不仅选择一个图片中最合适的区域，还可以把这些区域变换到一个标准的我们所期待的样子以简化后续层的推断。

空间变换可以被标准反向传播训练，允许包含这一模块的整个模型进行端到端的训练。

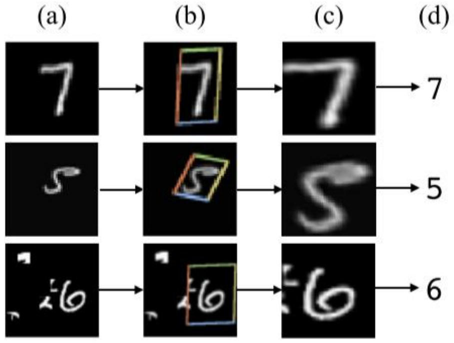
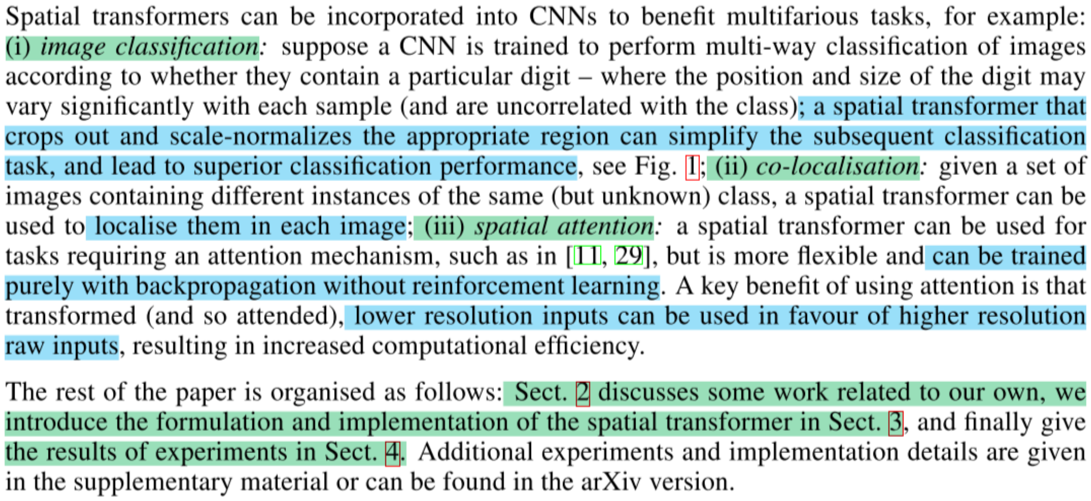


Figure 1: The result of using a spatial transformer as the first layer of a fully-connected network trained for distorted MNIST digit classification. (a) The input to the spatial transformer network is an image of an MNIST digit that is distorted with random translation, scale, rotation, and clutter. (b) The localisation network of the spatial transformer predicts a transformation to apply to the input image. (c) The output of the spatial transformer, after applying the transformation. (d) The classification prediction produced by the subsequent fully-connected network on the output of the spatial transformer. The spatial transformer network (a CNN including a spatial transformer module) is trained end-to-end with only class labels – no knowledge of the groundtruth transformations is given to the system.



Spatial transformers can be incorporated into CNNs to benefit multifarious tasks, for example: (i) *image classification*: suppose a CNN is trained to perform multi-way classification of images according to whether they contain a particular digit – where the position and size of the digit may vary significantly with each sample (and are uncorrelated with the class); a spatial transformer that crops out and scale-normalizes the appropriate region can simplify the subsequent classification task, and lead to superior classification performance, see Fig. 1; (ii) *co-localisation*: given a set of images containing different instances of the same (but unknown) class, a spatial transformer can be used to localise them in each image; (iii) *spatial attention*: a spatial transformer can be used for tasks requiring an attention mechanism, such as in [11, 29], but is more flexible and can be trained purely with backpropagation without reinforcement learning. A key benefit of using attention is that transformed (and so attended), lower resolution inputs can be used in favour of higher resolution raw inputs, resulting in increased computational efficiency.

The rest of the paper is organised as follows: Sect. 2 discusses some work related to our own, we introduce the formulation and implementation of the spatial transformer in Sect. 3, and finally give the results of experiments in Sect. 4. Additional experiments and implementation details are given in the supplementary material or can be found in the arXiv version.

空间变换网络插入 CNN 中，对多种任务均有益：

（1）image classification

空间变换可以把输入剪切并缩放至合适的区域以简化后续的分类任务，提升

分类性能。

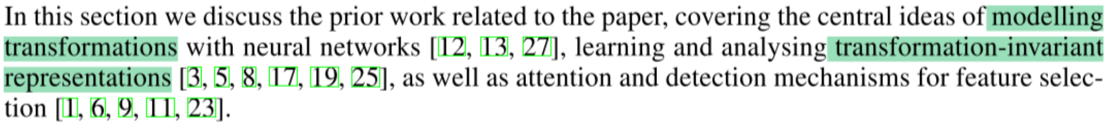
（2）co-localisation

给定一组包含一类物体不同实例的图像，空间变换器可以在每张图像中定位它们。

（3）spatial attention

空间变换器可以被用于需要 attention 机制的任务中，它更加灵活并且可以只用反向传播进行训练。

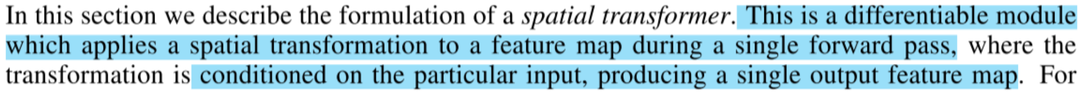
2. Related Work



In this section we discuss the prior work related to the paper, covering the central ideas of modelling transformations with neural networks [12, 13, 27], learning and analysing transformation-invariant representations [3, 5, 8, 17, 19, 25], as well as attention and detection mechanisms for feature selection [1, 6, 9, 11, 23].

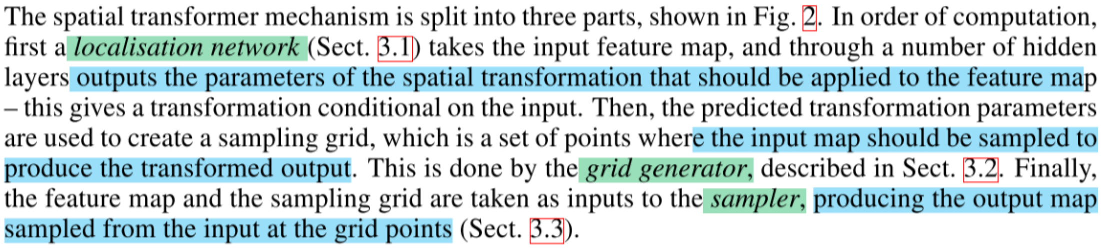
这一节是和本文有关的一些网络模型的讨论。

3. Spatial Transformers



In this section we describe the formulation of a *spatial transformer*. This is a differentiable module which applies a spatial transformation to a feature map during a single forward pass, where the transformation is conditioned on the particular input, producing a single output feature map. For

这是一个在单个前向传递中把空间变换应用到 feature map 上的可微模型，其中变换基于特定输入产生单个输出 feature map。



The spatial transformer mechanism is split into three parts, shown in Fig. 2. In order of computation, first a *localisation network* (Sect. 3.1) takes the input feature map, and through a number of hidden layers outputs the parameters of the spatial transformation that should be applied to the feature map – this gives a transformation conditional on the input. Then, the predicted transformation parameters are used to create a sampling grid, which is a set of points where the input map should be sampled to produce the transformed output. This is done by the *grid generator*, described in Sect. 3.2. Finally, the feature map and the sampling grid are taken as inputs to the *sampler*, producing the output map sampled from the input at the grid points (Sect. 3.3).

空间变换机制被分成三个部分：

localization network: 输入 feature map 通过一系列隐藏层后输出要被应用于 feature map 的空间变换参数。

grid generator: 根据变换参数确定在输入 feature map 和输出 feature map 上的映射关系。

sampler: 结合输入的 feature map 和映射关系，获得变换后的输出。

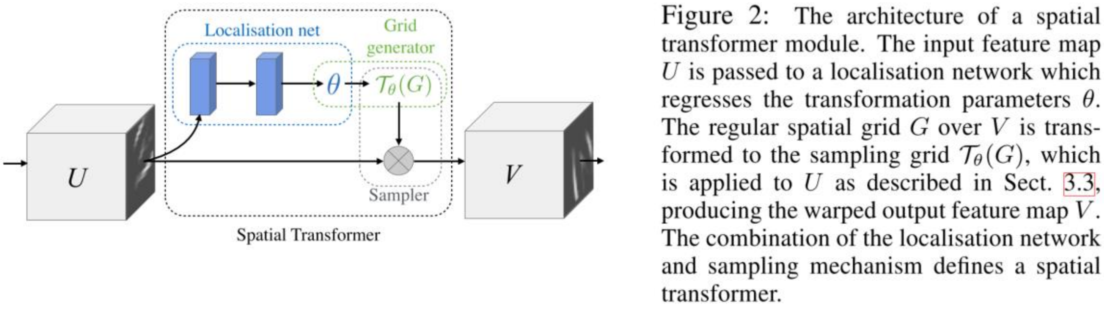
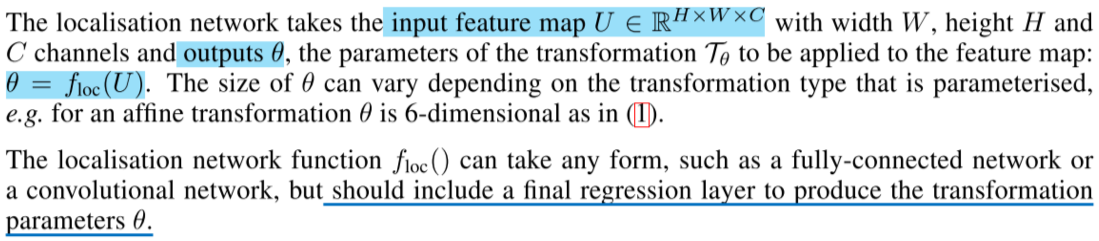


Figure 2: The architecture of a spatial transformer module. The input feature map U is passed to a localisation network which regresses the transformation parameters θ. The regular spatial grid G over V is transformed to the sampling grid $\mathcal{T}_\theta(G)$, which is applied to U as described in Sect. 3.3, producing the warped output feature map V. The combination of the localisation network and sampling mechanism defines a spatial transformer.

3.1 Localisation Network



The localisation network takes the input feature map $U \in \mathbb{R}^{H\times W\times C}$ with width W, height H and C channels and outputs θ, the parameters of the transformation $\mathcal{T}_\theta$ to be applied to the feature map: $\theta = f_{\text{loc}}(U)$. The size of θ can vary depending on the transformation type that is parameterised, *e.g.* for an affine transformation θ is 6-dimensional as in (1).

The localisation network function $f_{\text{loc}}()$ can take any form, such as a fully-connected network or a convolutional network, but should include a final regression layer to produce the transformation parameters θ.

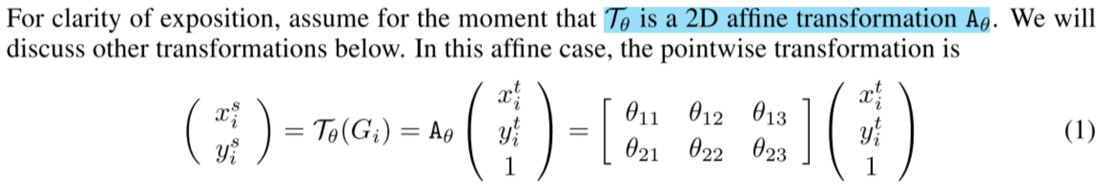
input: feature map $U \in R^{H\times W\times C}$（width W, height H, C channels）

output: $\theta = f_{loc}(U)$

localisation 网络函数$f_{loc}()$可以是任何形式，如全连接网络或卷积网络，但最后一定有一个回归层用于生成变换参数θ。

θ的形式可以根据需要而变化，以 2D 仿射变换为例，θ就是一个 2*3 的向量输出。

3.2 Parameterised Sampling Grid

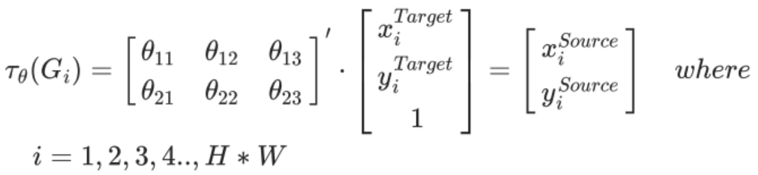


For clarity of exposition, assume for the moment that $\mathcal{T}_\theta$ is a 2D affine transformation $\mathtt{A}_\theta$. We will discuss other transformations below. In this affine case, the pointwise transformation is

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathtt{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{1}$$

考虑逆向仿射变换，先根据仿射变换输出的大小，生成输出的坐标网格点，再对该坐标位置矩阵中的点进行仿射变换。

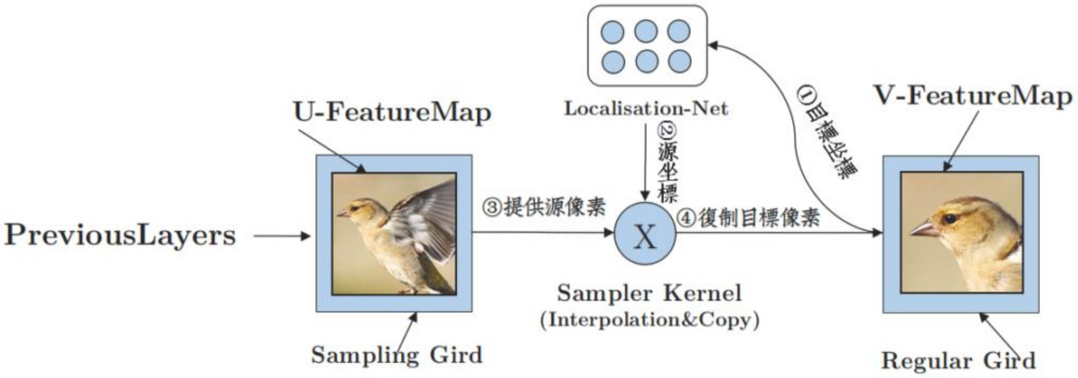
此时仿射系数为θ的逆矩阵，经仿射变换后可以得到 V 中的坐标点在 U 中的对应位置（可能非整数），再通过在输入图像中进行插值得到此坐标点的值。得到 U 中坐标点的值后，则可将其复制到 V 中，得到仿射变换结果。



$$\tau_\theta(G_i) = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix}' \cdot \begin{bmatrix} x_i^{Target} \\ y_i^{Target} \\ 1 \end{bmatrix} = \begin{bmatrix} x_i^{Source} \\ y_i^{Source} \end{bmatrix} \quad where$$

$$i = 1, 2, 3, 4.., H * W$$

具体过程参考下图：



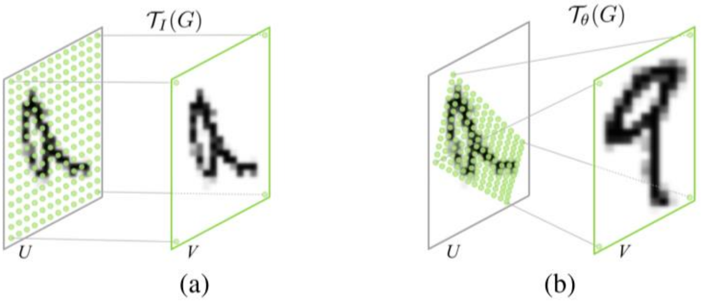


Figure 3: Two examples of applying the parameterised sampling grid to an image U producing the output V. (a) The sampling grid is the regular grid $G = \mathcal{T}_I(G)$, where I is the identity transformation parameters. (b) The sampling grid is the result of warping the regular grid with an affine transformation $\mathcal{T}_\theta(G)$.

3.3 Differentiable Image Sampling

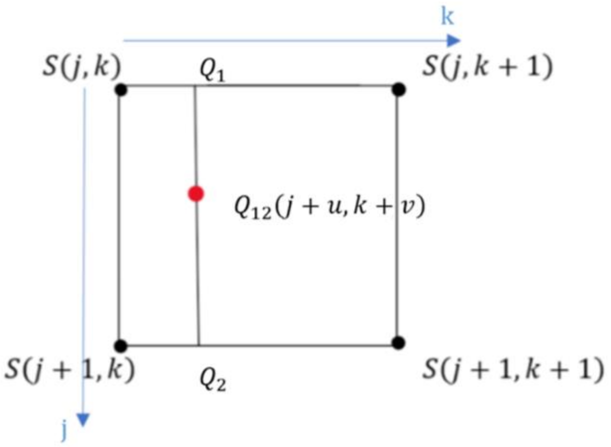
对双线性插值的情况有插值等式如下：

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

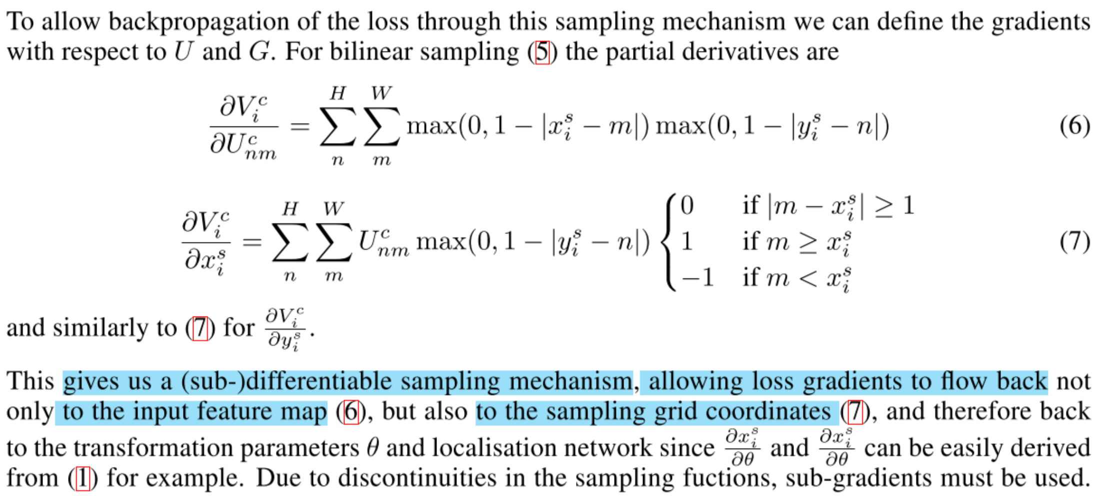
V 中(x_i^t, y_i^t)变换到 U 中(x_i^s, y_i^s)，选取(x_i^s, y_i^s)在 U 中的邻近点求和。

Q: 此处不理解，按照双线性插值方法直接选择临近的四个点进行插值就可以了，而这个式子好像是循环选邻近点再求和。

双线性插值：



$$Q_1 = s(j,k)(1-v) + s(j,k+1)v$$
$$Q_2 = S(j+1,k)(1-v) + S(j+1,k+1)v$$
$$Q_{12} = Q_1(1-u) + Q_2 u$$



To allow backpropagation of the loss through this sampling mechanism we can define the gradients with respect to U and G. For bilinear sampling (5) the partial derivatives are

$$\frac{\partial V_i^c}{\partial U_{nm}^c} = \sum_n^H \sum_m^W \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|) \tag{6}$$

$$\frac{\partial V_i^c}{\partial x_i^s} = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |y_i^s - n|) \begin{cases} 0 & \text{if } |m - x_i^s| \geq 1 \\ 1 & \text{if } m \geq x_i^s \\ -1 & \text{if } m < x_i^s \end{cases} \tag{7}$$

and similarly to (7) for $\frac{\partial V_i^c}{\partial y_i^s}$.

This gives us a (sub-)differentiable sampling mechanism, allowing loss gradients to flow back not only to the input feature map (6), but also to the sampling grid coordinates (7), and therefore back to the transformation parameters θ and localisation network since $\frac{\partial x_i^s}{\partial \theta}$ and $\frac{\partial x_i^s}{\partial \theta}$ can be easily derived from (1) for example. Due to discontinuities in the sampling fuctions, sub-gradients must be used.

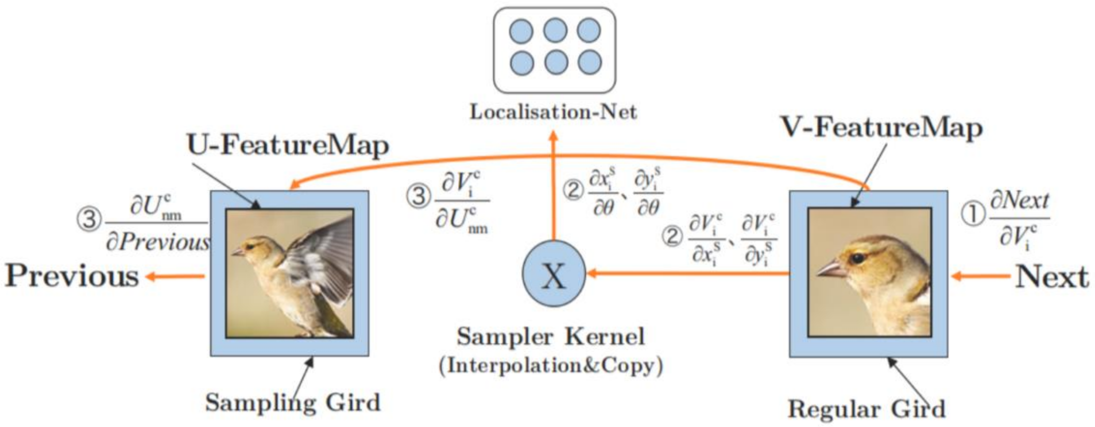
此处的采样核函数是不连续的，不能直接如下求导：

$$g = \frac{\partial V_i^C}{\partial \theta}$$

应该分两步，先对x_i^s和x_i^s求局部梯度$\frac{\partial V_i^C}{\partial x_i^s}$、$\frac{\partial V_i^C}{\partial y_i^s}$，后有：

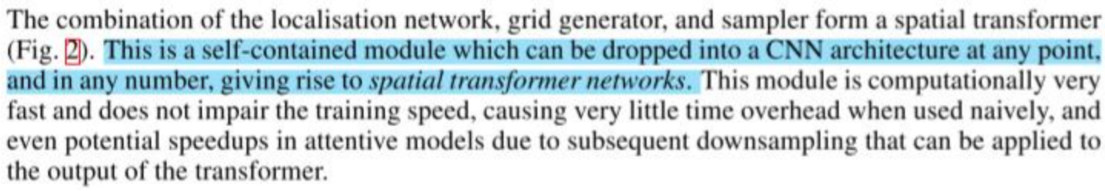
$$\begin{cases} g = \frac{\partial V_i^C}{\partial x_i^s} \cdot \frac{\partial x_i^s}{\partial \theta} \\ g = \frac{\partial V_i^C}{\partial y_i^s} \cdot \frac{\partial y_i^s}{\partial \theta} \end{cases}$$

梯度流动过程参考下图：


Localisation-Net
U-FeatureMap
V-FeatureMap
③ $\frac{\partial U_{nm}^c}{\partial Previous}$
Previous
③ $\frac{\partial V_i^c}{\partial U_{nm}^c}$
② $\frac{\partial x_i^s}{\partial \theta}$、$\frac{\partial y_i^s}{\partial \theta}$
② $\frac{\partial V_i^c}{\partial x_i^s}$、$\frac{\partial V_i^c}{\partial y_i^s}$
① $\frac{\partial Next}{\partial V_i^c}$
Next
X
Sampler Kernel
(Interpolation&Copy)
Sampling Gird
Regular Gird

★★★空間變換層反向傳播梯度流動，①②③代表分支流

3.4 Spatial Transformer Networks



The combination of the localisation network, grid generator, and sampler form a spatial transformer (Fig. 2). This is a self-contained module which can be dropped into a CNN architecture at any point, and in any number, giving rise to *spatial transformer networks*. This module is computationally very fast and does not impair the training speed, causing very little time overhead when used naively, and even potential speedups in attentive models due to subsequent downsampling that can be applied to the output of the transformer.

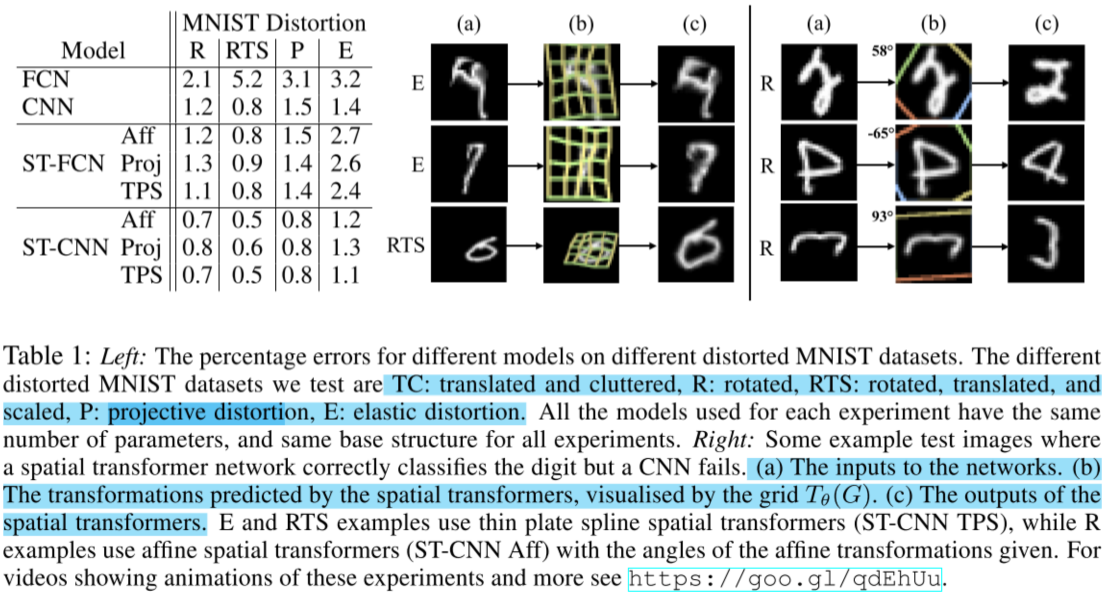
空间变换器是一个独立的模块，可以在任何时候放入 CNN 架构中，产生空间变

换网络。这个模块的计算速度很快，不会损害训练速度。

4. Experiments

4.1 Distorted MNIST

本节中使用 MNIST 手写数据集作为测试平台，训练不同的神经网络模型来分类以方式失真的 MNIST 数据。使用空间转换的网络得到的结果优于其对应的基础网络。



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

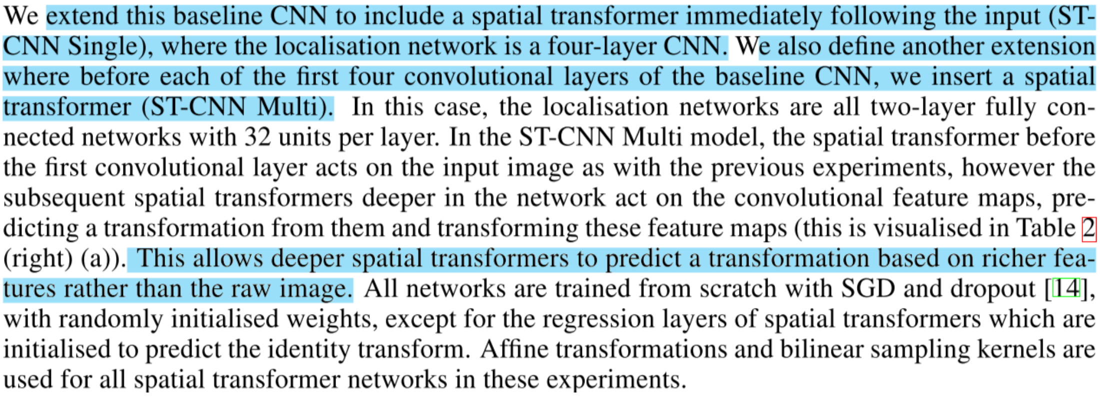
(a)
(b)
(c)
E
E
RTS
(a)
(b)
(c)
R
R
R
58°
-65°
93°

Table 1: *Left:* The percentage errors for different models on different distorted MNIST datasets. The different distorted MNIST datasets we test are TC: translated and cluttered, R: rotated, RTS: rotated, translated, and scaled, P: projective distortion, E: elastic distortion. All the models used for each experiment have the same number of parameters, and same base structure for all experiments. *Right:* Some example test images where a spatial transformer network correctly classifies the digit but a CNN fails. (a) The inputs to the networks. (b) The transformations predicted by the spatial transformers, visualised by the grid $\mathcal{T}_\theta(G)$. (c) The outputs of the spatial transformers. E and RTS examples use thin plate spline spatial transformers (ST-CNN TPS), while R examples use affine spatial transformers (ST-CNN Aff) with the angles of the affine transformations given. For videos showing animations of these experiments and more see https://goo.gl/qdEhUu.

上表展示了不同模型在不同畸变的 MNIST 数据集下的错误率结果。
对数据集进行畸变的方式有多种，插入的空间变换网络也使用了多种变换方式。
可以看出，加入空间变换网络之后取得了更低的错误率。

4.2 Street View House Numbers

现在在一个具有挑战性的现实世界数据集 Street View House Numbers（SVHN）上测试我们的空间变换网络。该数据集包含大约 200k 个房屋号码的真实世界图像，其任务是识别每个图像中的数字序列。 每幅图像中有 1 到 5 位数字，规模和空间布局变化很大。



We extend this baseline CNN to include a spatial transformer immediately following the input (ST-CNN Single), where the localisation network is a four-layer CNN. We also define another extension where before each of the first four convolutional layers of the baseline CNN, we insert a spatial transformer (ST-CNN Multi). In this case, the localisation networks are all two-layer fully connected networks with 32 units per layer. In the ST-CNN Multi model, the spatial transformer before the first convolutional layer acts on the input image as with the previous experiments, however the subsequent spatial transformers deeper in the network act on the convolutional feature maps, predicting a transformation from them and transforming these feature maps (this is visualised in Table 2 (right) (a)). This allows deeper spatial transformers to predict a transformation based on richer features rather than the raw image. All networks are trained from scratch with SGD and dropout [14], with randomly initialised weights, except for the regression layers of spatial transformers which are initialised to predict the identity transform. Affine transformations and bilinear sampling kernels are used for all spatial transformer networks in these experiments.

ST-CNN Single: 将 baseline CNN 扩展为在输入后立即包含一个空间变换器。
ST-CNN Multi: 在 baseline CNN 的前四个卷积层之前，各插入一个空间变换器。
在 ST-CNN Multi 中，第一卷积层之前的空间变换器如同前面的实验一样作用于

输入图像，然而随后的网络中较深的空间变换器作用于卷积特征映射，预测它们的变换和变换这些特征图。这允许更深的空间变换器基于更丰富的特征而不是原始图像来预测变换。

Model		Size 64px	 128px
Maxout CNN [10]		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**

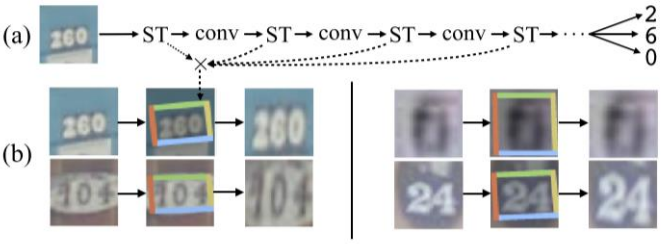


Table 2: *Left:* The sequence error (%) for SVHN multi-digit recognition on crops of 64 × 64 pixels (64px), and inflated crops of 128 × 128 (128px) which include more background. *The best reported result from [1] uses model averaging and Monte Carlo averaging, whereas the results from other models are from a single forward pass of a single model. *Right:* (a) The schematic of the ST-CNN Multi model. The transformations of each spatial transformer (ST) are applied to the convolutional feature map produced by the previous layer. (b) The result of the composition of the affine transformations predicted by the four spatial transformers in ST-CNN Multi, visualised on the input image.

可见加入空间变换器的网络取得了更好的结果。

4.3 Fine-Grained Classification

In this section, we use a spatial transformer network with multiple transformers in parallel to perform fine-grained bird classification. We evaluate our models on the CUB-200-2011 birds dataset [28], containing 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, and require detailed texture and shape analysis to distinguish. In our experiments, we only use image class labels for training.

在本节中，我们使用具有多个变换器的的空间变换器网络并行执行细粒度鸟类分类。

Model		
Cimpoi '15 [4]		66.7
Zhang '14 [30]		74.9
Branson '14 [2]		75.7
Lin '15 [20]		80.9
Simon '15 [24]		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**

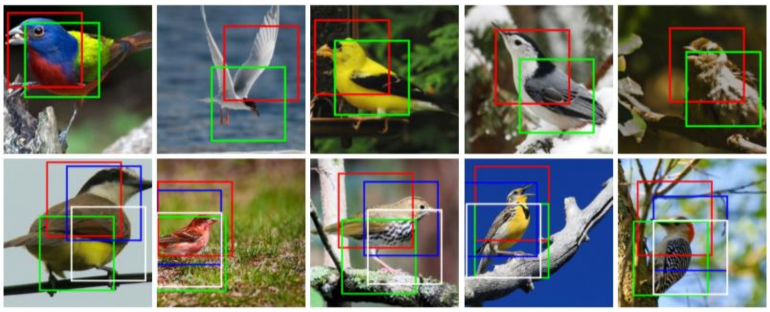
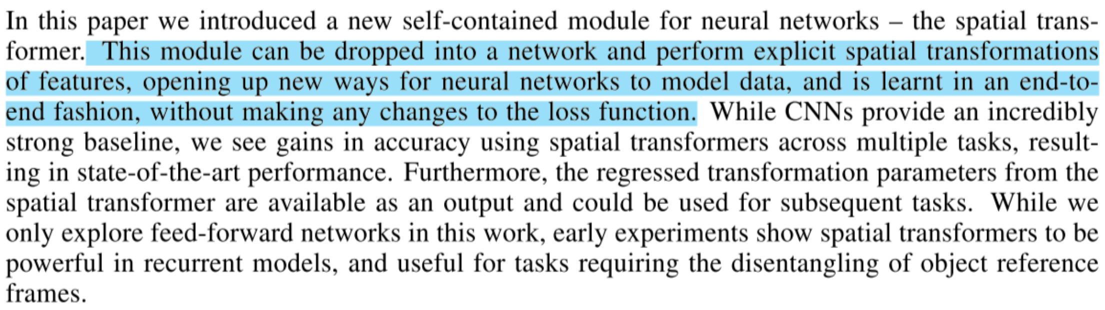


Table 3: *Left:* The accuracy (%) on CUB-200-2011 bird classification dataset. Spatial transformer networks with two spatial transformers (2×ST-CNN) and four spatial transformers (4×ST-CNN) in parallel outperform other models. 448px resolution images can be used with the ST-CNN without an increase in computational cost due to downsampling to 224px *after* the transformers. *Right:* The transformation predicted by the spatial transformers of 2×ST-CNN (top row) and 4×ST-CNN (bottom row) on the input image. Notably for the 2×ST-CNN, one of the transformers (shown in red) learns to detect heads, while the other (shown in green) detects the body, and similarly for the 4×ST-CNN.

具有 4 个并行空间变换器的空间变换网络有最高的准确率。

5. Conclusion



In this paper we introduced a new self-contained module for neural networks – the spatial transformer. This module can be dropped into a network and perform explicit spatial transformations of features, opening up new ways for neural networks to model data, and is learnt in an end-to-end fashion, without making any changes to the loss function. While CNNs provide an incredibly strong baseline, we see gains in accuracy using spatial transformers across multiple tasks, resulting in state-of-the-art performance. Furthermore, the regressed transformation parameters from the spatial transformer are available as an output and could be used for subsequent tasks. While we only explore feed-forward networks in this work, early experiments show spatial transformers to be powerful in recurrent models, and useful for tasks requiring the disentangling of object reference frames.

本文介绍了一种用于神经网络的新型独立模块：空间变换器。

这个模块可以放入网络中执行特征的显式空间转换，为神经网络建模数据开辟了新的途径，并以端到端的方式学习，而不对损失函数做任何改变。

使用跨多任务的空间变换器可以提高准确度，从而获得最先进的性能。

来自空间转换器的回归变换参数可用作输出，并可用于后续任务。