# Spatial Transformer Networks

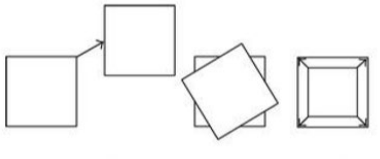
Jaderberg, Max, Karen Simonyan, and Andrew Zisserman.
Advances in neural information processing systems 28 (2015).

2022.11.11

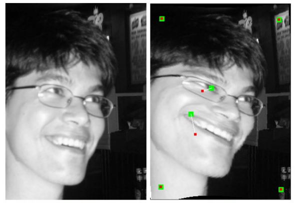
Advanced Computer Vision

KENTECH Energy AI
VIEW Lab
Sohee Kim

Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).




# ◆ Introduction




- The Convolutional Neural Network (CNN) define an exceptionally powerful class of models,
  but has Limitation : **lack of ability to be <u>Spatially invariant</u> to the input data**
  == invariance of the model towards spatial transformations of images (rotation, translation, scaling)
- <u>Max-pooling layer</u> – satisfy this, but 2x2 pixel-wise operation is difficult to cope with various spatial variability
  ⇒ **Spatial transformer module** – include in to standard neural network


**Spatial transformer** module

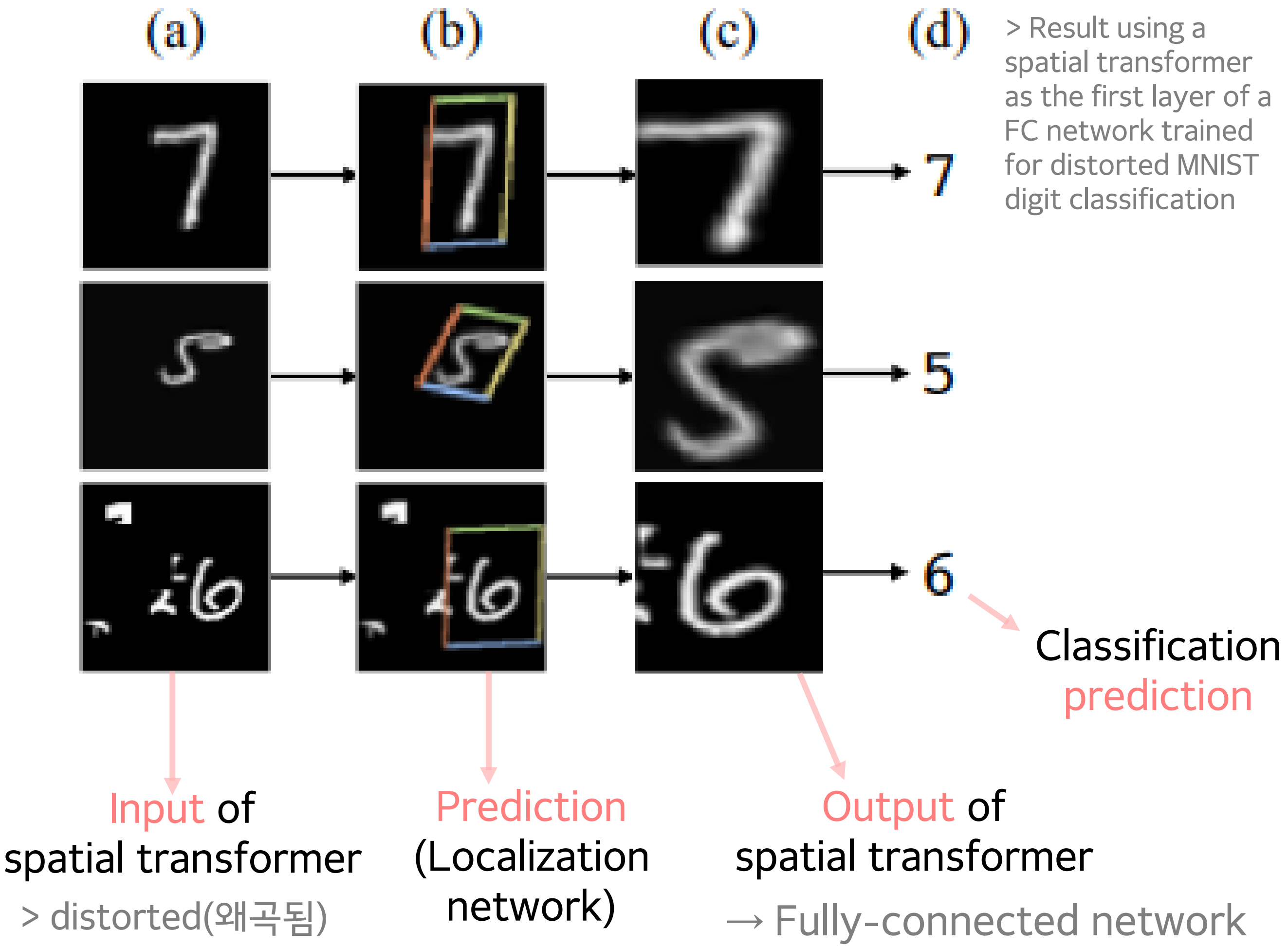
- A learnable module that can be placed in a CNN, to increase the spatial invariance in an efficient manner
- A dynamic mechanism – unlike pooling layer (receptive fields are fixed and local)
- Input image → transformation is performed on the entire feature map (can include scaling, cropping, rotations, non-rigid deformations)
- Networks select regions that are most relevant(attention), transform those regions to a canonical(일반적인) pose

# ◆ Introduction

❖ **Image classification**

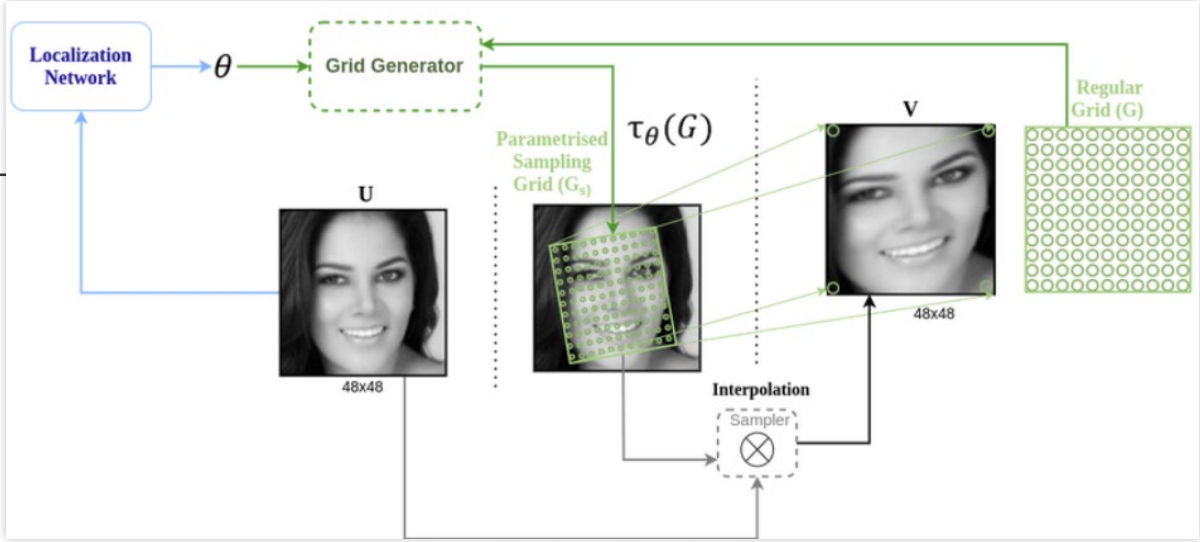
**Using Spatial Transformer**

- The action of the spatial transformer is conditioned on individual data samples, trained well without extra supervision
- Can be trained with standard back-propagation (allowing **end-to-end** training of models)
- Can be added into CNNs to help with a variety of tasks (image classification, co-localization, spatial attention)
- The framework they present in this paper can be seen as a generalisation of differentiable attention to any spatial transformation



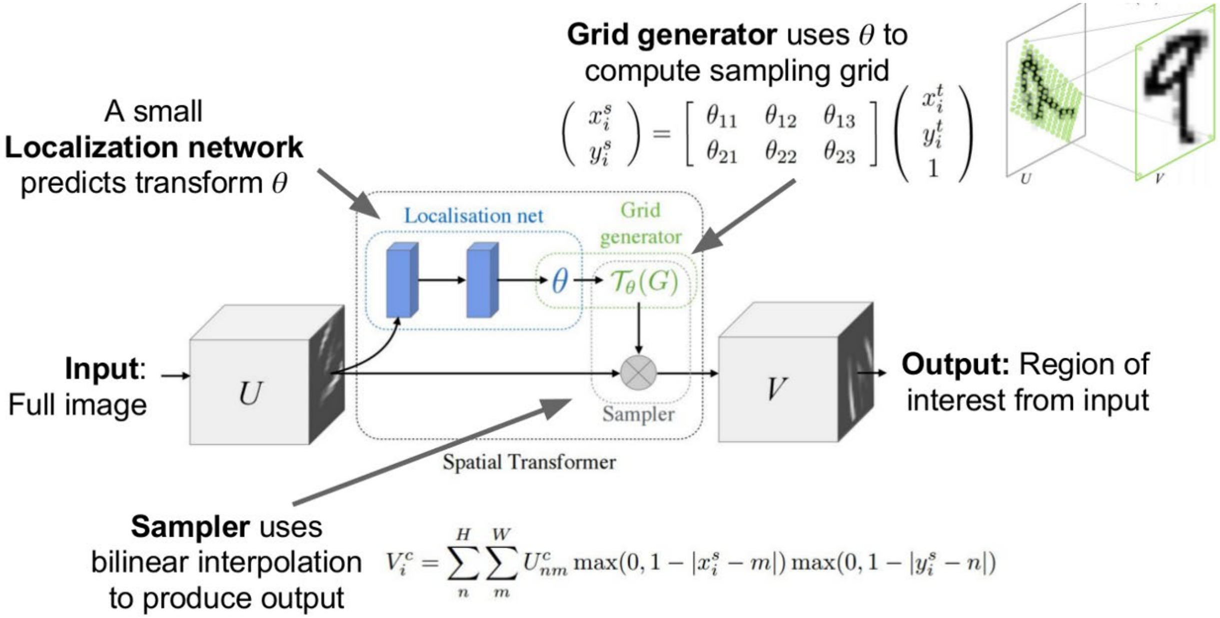
Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).

# ◆ Spatial Transformers




## ❖ Spatial Transformer

1. Localisation network
2. Grid generator
3. Sampler




(W x H x Ch x Batch size)

1. Input feature map U ⇒ **localisation net**

   ⇒ Transformation parameter $\theta$

2. $\theta$ ⇒ **Grid generator** ⇒ Sampling grid $T_\theta(G)$

   (include sampling point location)

3. U + $T_\theta(G)$ ⇒ **Sampler**

   → $T_\theta(G)$ has sampling point – apply to U

   ⇒ Output feature map V

https://manjunathbhat9920.medium.com/spatial-transformer-network-82666f184299

Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).

# ◆ Spatial Transformers

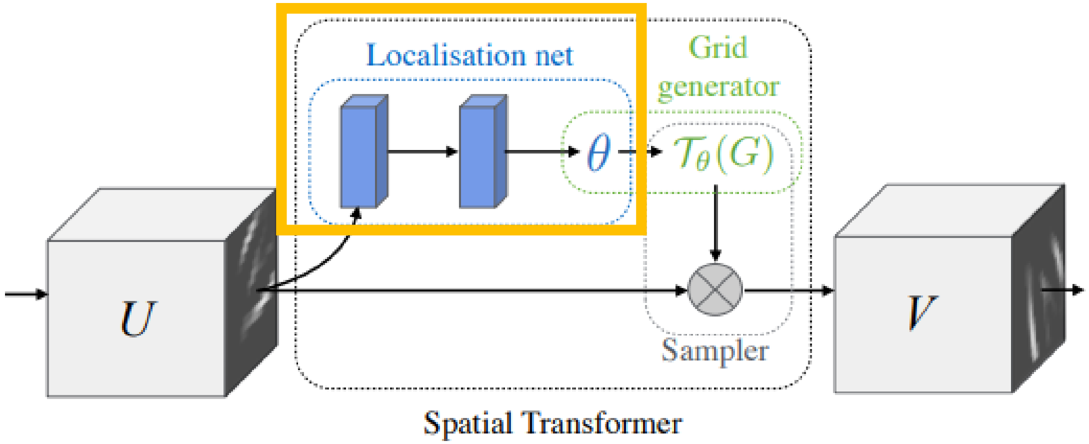
## 1. Localisation network

$$A_\theta = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix}$$

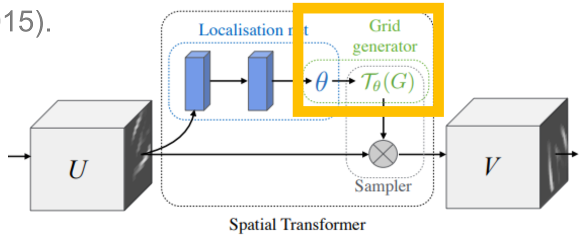
> Only scaling + translation

- The localisation network is composed of fully-connected layers or convolution layers
- Predict **Transformation parameter matrix $\theta$** as output
- Input : $U \in R^{H \times W \times C}$ (width W, height H and C channels) => $\theta = f_{loc}(U)$
- The size of $\theta$ vary depending on transformation type (ex. affine=6-dimensional)
- Localisation network function $f_{loc}(U)$ can take any form (Convolution layer, FC layer) but Must include a final regression layer to produce the transform parameters.

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \underbrace{\begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix}}_{\text{Transformation parameter matrix}} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).



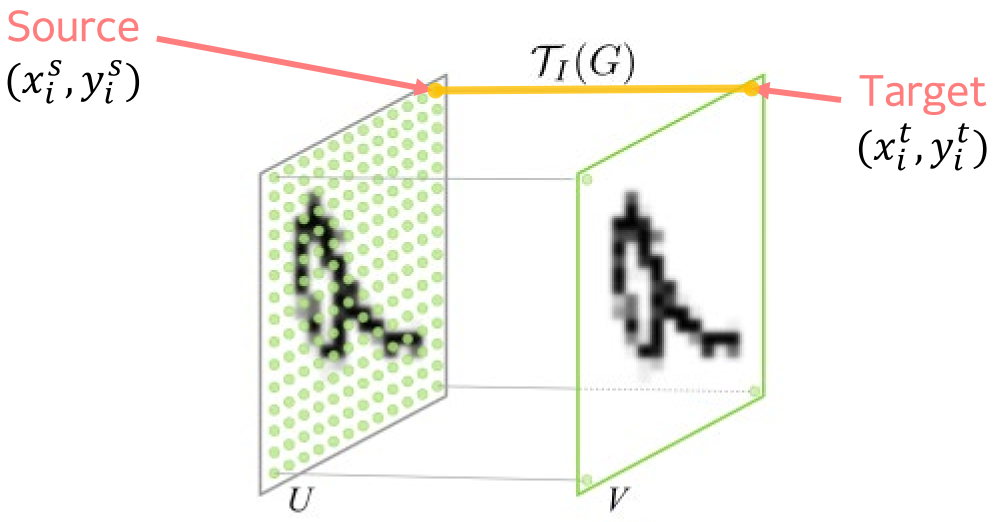

# ◆ Spatial Transformers

## 2. Parameterised Sampling Grid

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

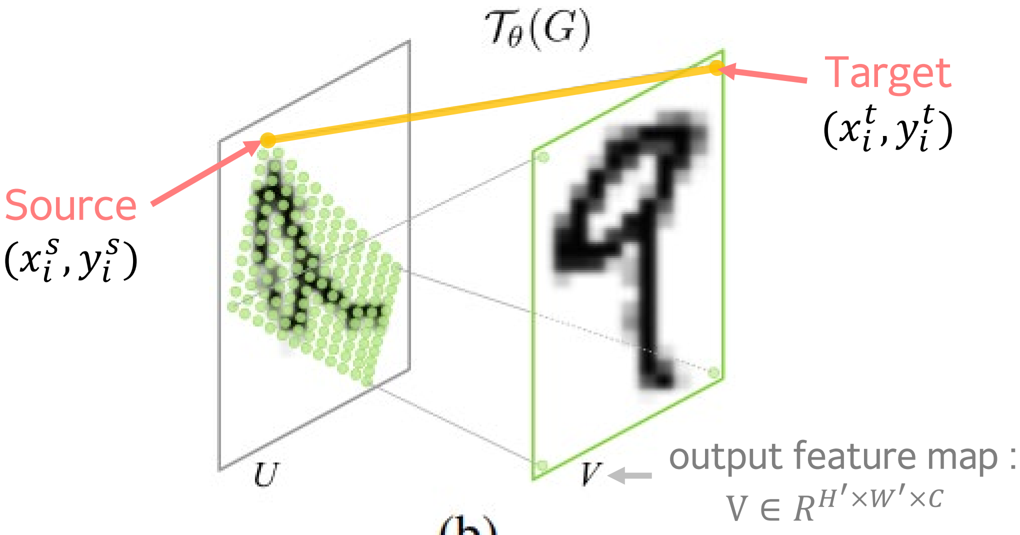
- Regular grid G = $(x_i^t, y_i^t)$ ⇒ Apply transformation $T_\theta$ on G ⇒ $\boldsymbol{T_\theta(G)}$
- **Sampling Grid $\boldsymbol{T_\theta(G)}$** = result of warping the regular grid G with transformation $T_\theta$

✓ Grid generator – **Identity transformation**




(a) Sampling Grid :
Regular grid $G = T_I(G)$

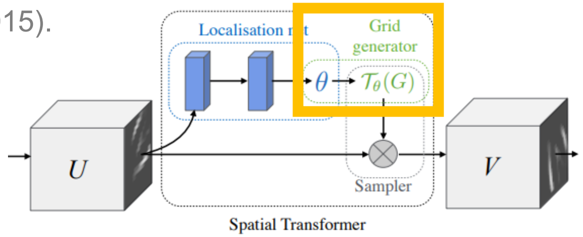
✓ Grid generator – **Affine transformation**




(b)

Sampling Grid : $G = T_\theta(G)$

Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).




# ◆ Spatial Transformers

## 2. Parameterised Sampling Grid

✓ **Grid generator Examples**

➢ The transformation can have **any parameterised form**, provided that it is **differentiable** with respect to the parameters

### Affine transform

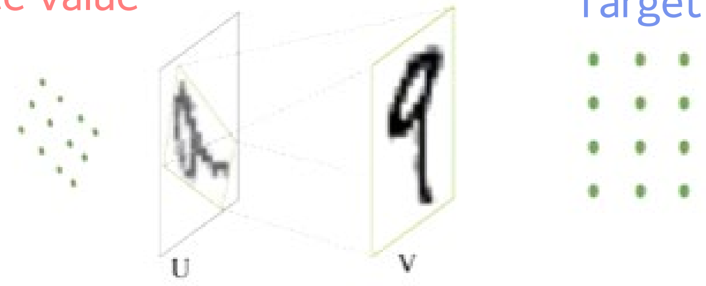
scale, rotation, translation, skew, cropping

Affine Transformation

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

Source value

Target value

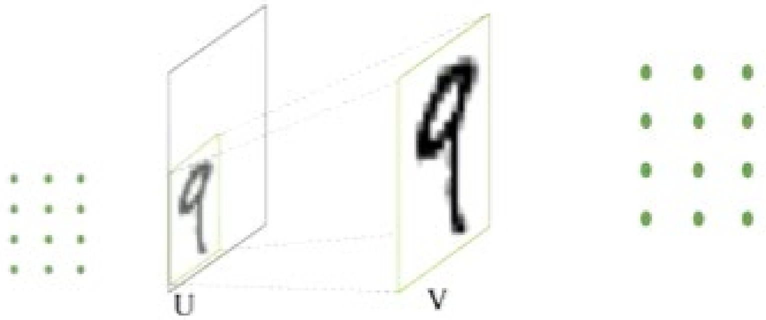



### Attention model

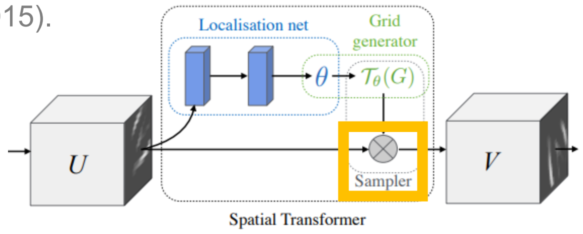
isotropic scale, translation, cropping

Attention Transformation

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathtt{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \boxed{\begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix}} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



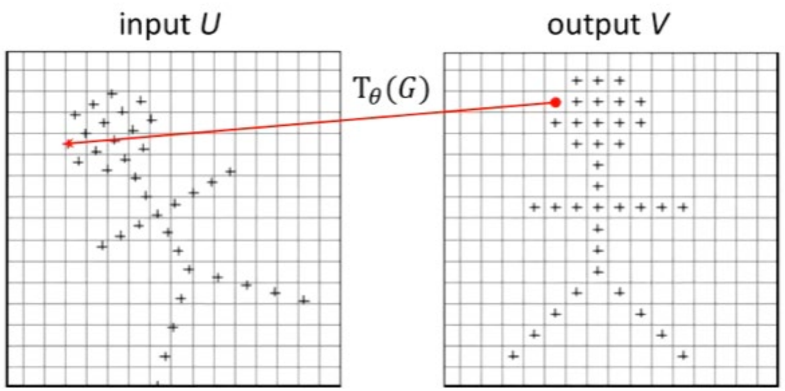
Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).




## ◆ Spatial Transformers

### 3. Differentiable Image Sampling

- $Sample(T_\theta(G), U) \rightarrow \boldsymbol{V}$
- Sampler : the set of sampling points $\boldsymbol{T_\theta(G)}$ + input feature map U ⇒ produce sampled **output feature map** V
- Each $(x_i^s, y_i^s)$ coordinate in $T_\theta(G)$ defines the spatial location in the input U where sampling kernel is applied to get the value at a particular pixel in the output V




- With high probability, location points are not an exact integer (like (1, 2))
  → need **interpolation** of surrounding values
- Any sampling kernel can be used.
- Interpolation function: Sampling kernel k()

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y) \ \forall i \in [1 \dots H'W'] \ \forall c \in [1 \dots C]$$

Parameters of k()

Output value $(x_i^t, y_i^t)$
Output feature map channel c

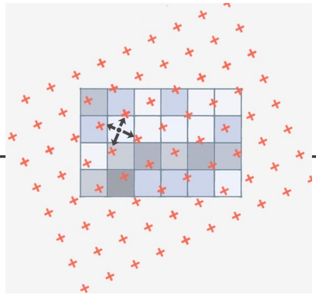
Input value
(n,m) value

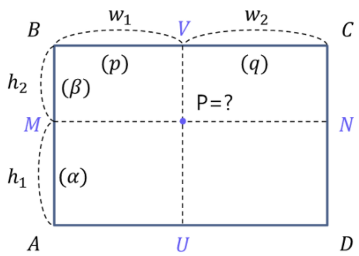
Sampling grid coordinate

> General form

https://jamiekang.github.io/2017/05/27/spatial-transformer-networks/

Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).






# ◆ Spatial Transformers

## 3. Differentiable Image Sampling

### ▪ Nearest Integer Sampling

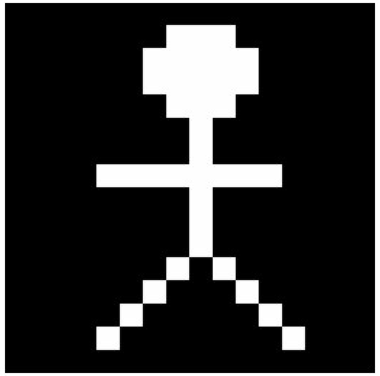
$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c \delta(\lfloor x_i^s + 0.5 \rfloor - m)\delta(\lfloor y_i^s + 0.5 \rfloor - n)$$

Target value ($V_i^c$)

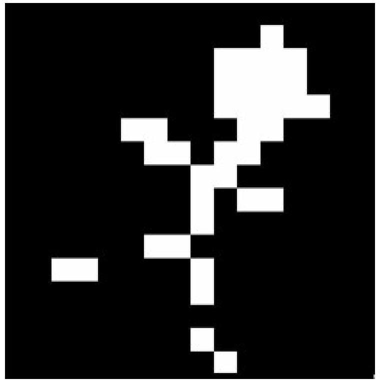
Source value ($U_{nm}^c$)

[ ] : rounding to the nearset integer

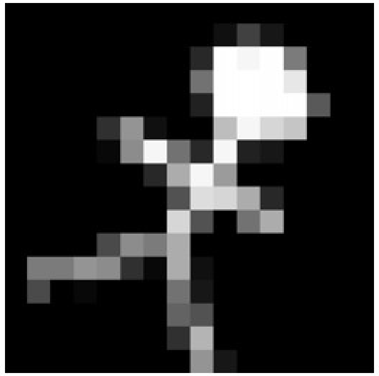
Original Image



Nearest Neighbor



Bilinear Interpolation



### ▪ Bilinear Sampling → linear interpolation on the x, y axes respectively

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

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

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

- ➢ It is a (sub-)**differentiable** sampling mechanism so that it is convenient for backpropagation
- ➢ If non-differentiable, backpropagation can be calculated by dividing it by sections.

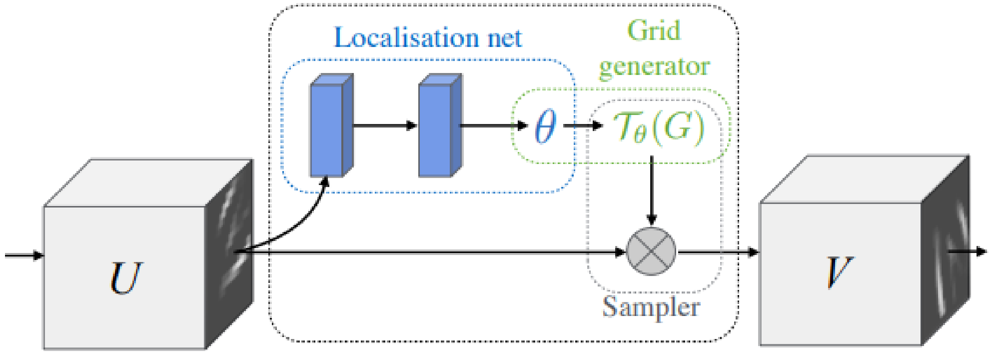
Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).

# ◆ Spatial Transformers

## ❖ Spatial Transformer Networks

- The combination of Localisation network, grid generator and sampler form spatial transformer module

  + CNN achitecture ⇒ **Spatial Transformer Network**
- Spatial transformer module can be dropped into a CNN architecture at any point, and in any number
- Learn during the training process to minimize the overall cost function of CNN ⇒ it has little effect on training speed
- The knowledge(how to transform) is compressed and cached in the weights of the localisation network during training
- It is possible to have multiple spatial transformers in a CNN.
  - To put the ST layer just before the CNN input : most efficient

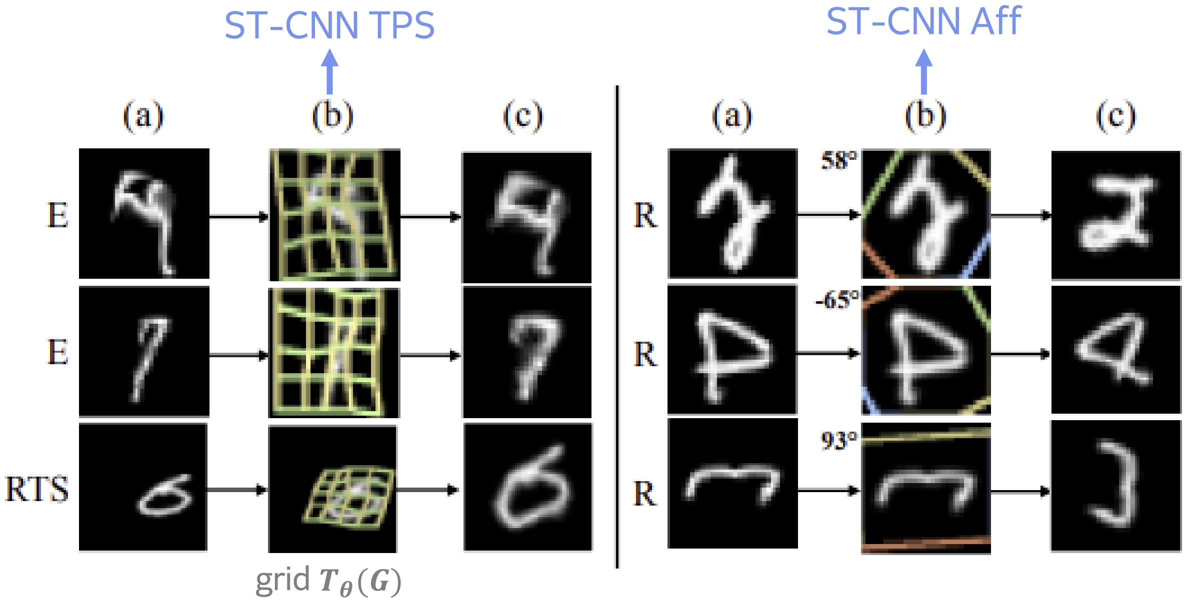


# ◆ Experiments

## 1. Distorted MNIST

✓ Model
- FCN, CNN
- ST + FCN
- ST + CNN

| 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 |



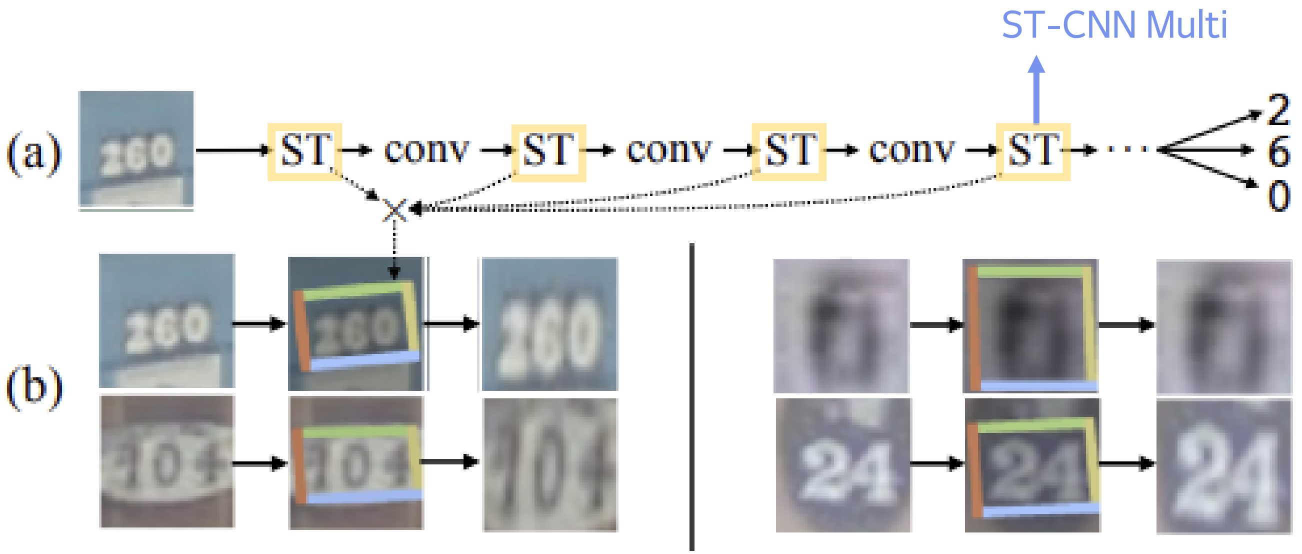

- MNIST dataset – distorted in various ways (4)
  - rotation (R), rotation-translation-scale (RTS), projective transformation (P), elastic warping (E)
- As we can see, ST-FCN outperforms FCN and ST-CNN outperforms CNN.
  - ST-CNN models consistently perform better than ST-FCN models
- CNN > FCN ⇒ The CNN models include two max-pooling layers → Maxpooling layer makes spatial invariance high
- Spatial transformation : affine (AFF), projective (Proj), thin plate spline transformation (TPS)
  - Between different classes of transformation, the TPS is the most powerful

Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).

## ◆ Experiments

### 2. Street View House Numbers

| Model | | Size 64px | Size 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** |
| ST-CNN | Multi | **3.6** | **3.9** |



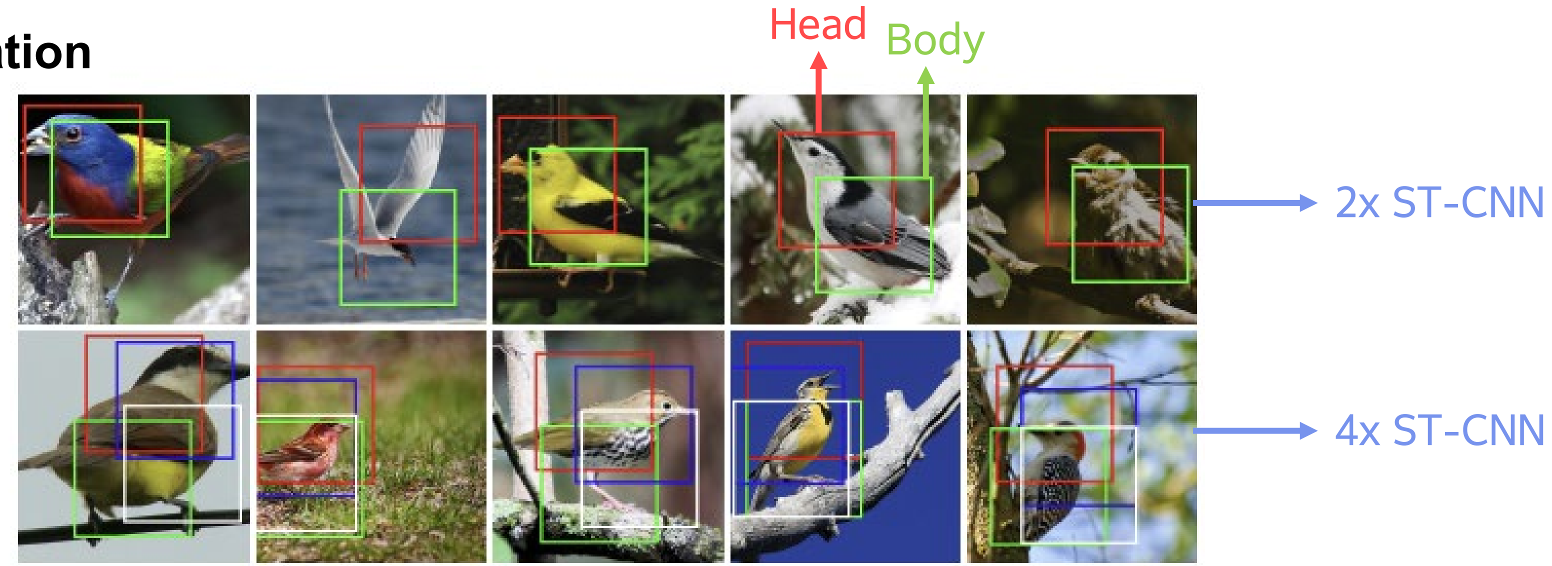

- This dataset contains around 200k real world images of house numbers
- Put ST layer → CNN convolutional stack
  - ST-CNN Single : Only one ST at the beginning of network
  - ST-CNN Multi : One ST before each convolutional layer
- Affine transformation, bilinear sampler is used
- ST-CNN outperforms Maxout and CNN
  - ST-CNN Multi outperforms ST-CNN Single a bit– 속도 6% 느려짐

Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).

# ◆ Experiments

## 3. Fine-Grained Classification

| 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** |

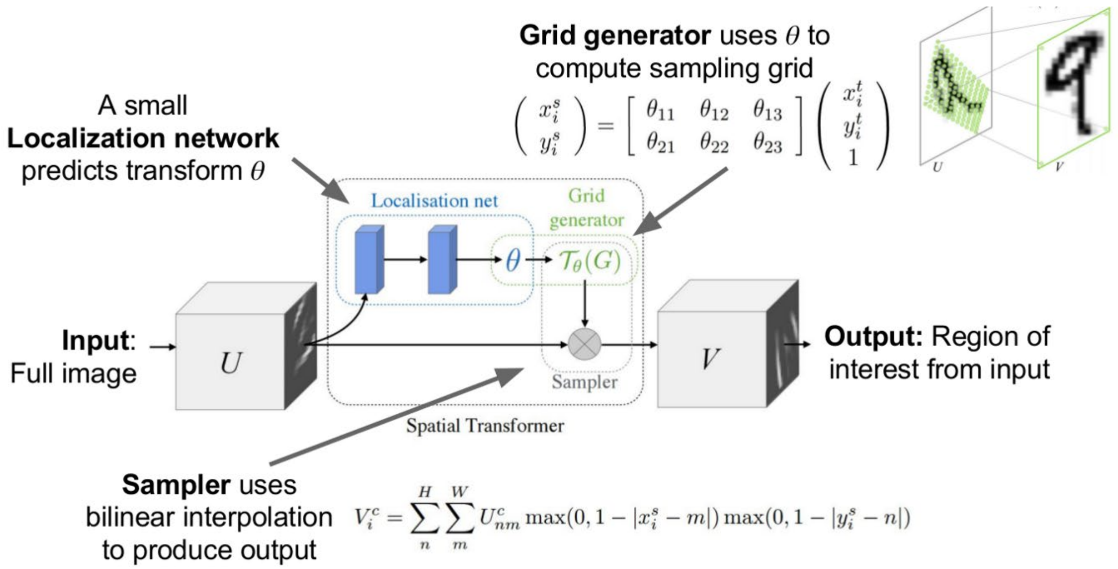


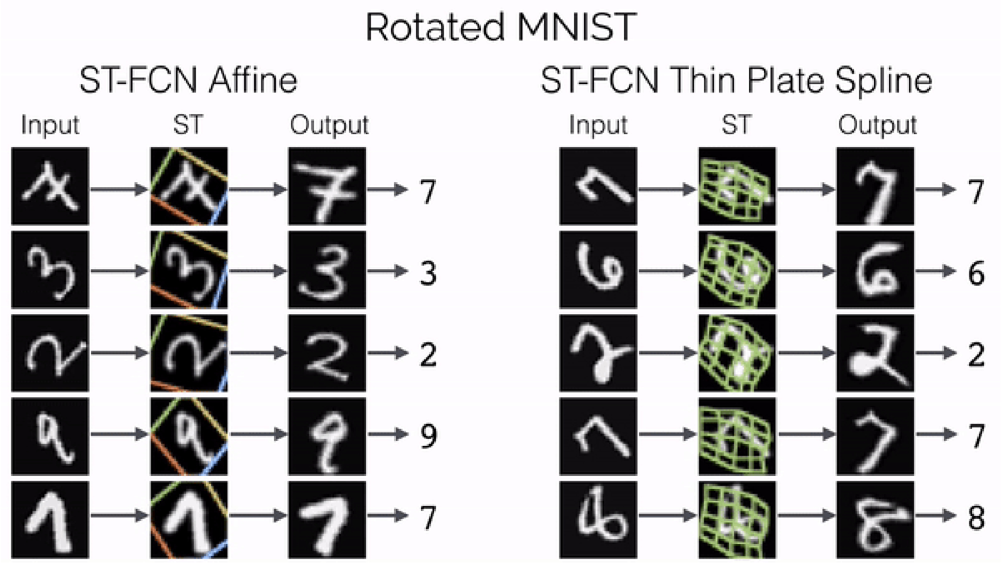

- Use a spatial transformer network with multiple transformers in parallel to perform fine-grained bird classification
- CUB 200-2011 birds dataset - 200 species of birds, 11,788 images
- Strong baseline CNN model : Inception-v2 (ImageNet pre-trained) for classifying 200 species (82.3% accuracy)
  - 2x ST-CNN , 4x ST-CNN : 2 or 4 parallel STs
- 4x ST-CNN achieves an accuracy of 84.1%, outperforms the baseline by 1.8%
- Interesting behaviour : Each box = ST found (without supervision)
  - 2x ST-CNN : red ST – head detector, green – detect central part of the body

Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).

# ◆ Conclusion

- 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
- **Differentiable** and learnt in an end-to-end fashion – without making any changes to the loss function
- While CNNs provide an incredibly strong baseline, they see gains in accuracy using spatial transformers across multiple tasks, resulting in state-of-the-art performance.




https://manjunathbhat9920.medium.com/spatial-transformer-network-82666f184299
https://towardsdatascience.com/review-stn-spatial-transformer-network-image-classification-d3cbd98a70aa