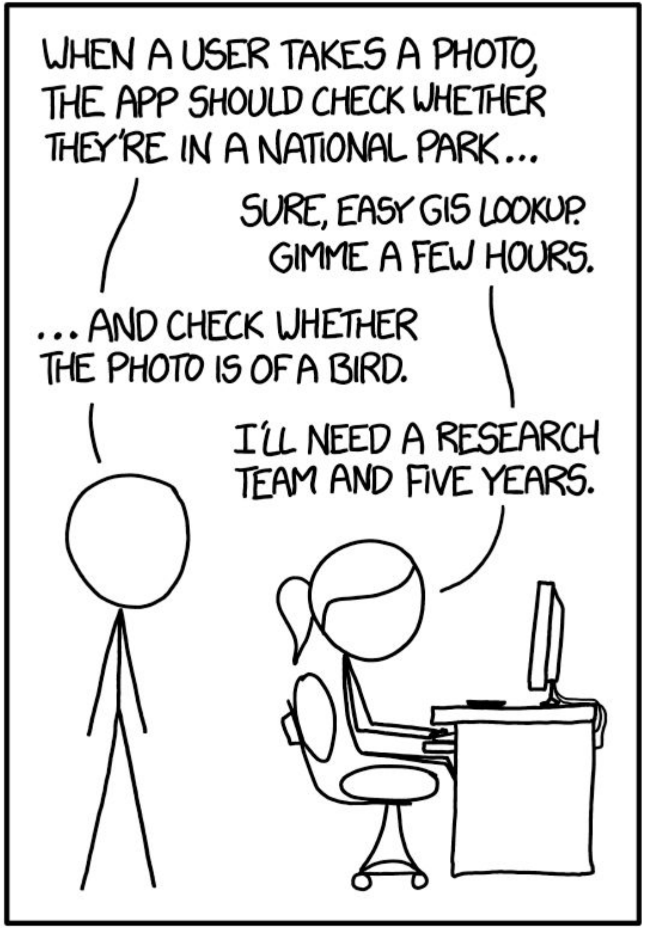
Deep Learning meets Spatial Transformations vol. 1

Tomáš Karella

Outline

- Motivation
- Architecture
- Experiments

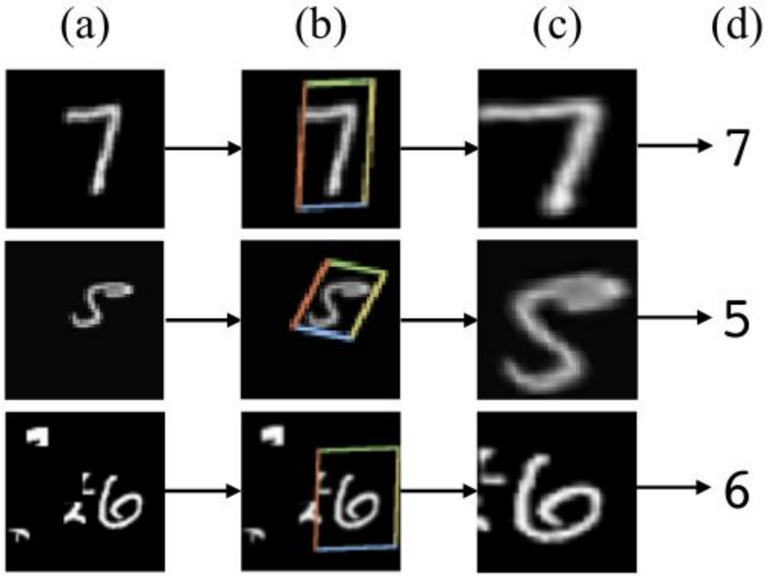
JADERBERG, Max, et al. Spatial transformer networks. *Advances in neural information processing systems*, 2015, 28.



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

Motivation

- “Invariant” in CNN models
- “Competitive” on tiny HW



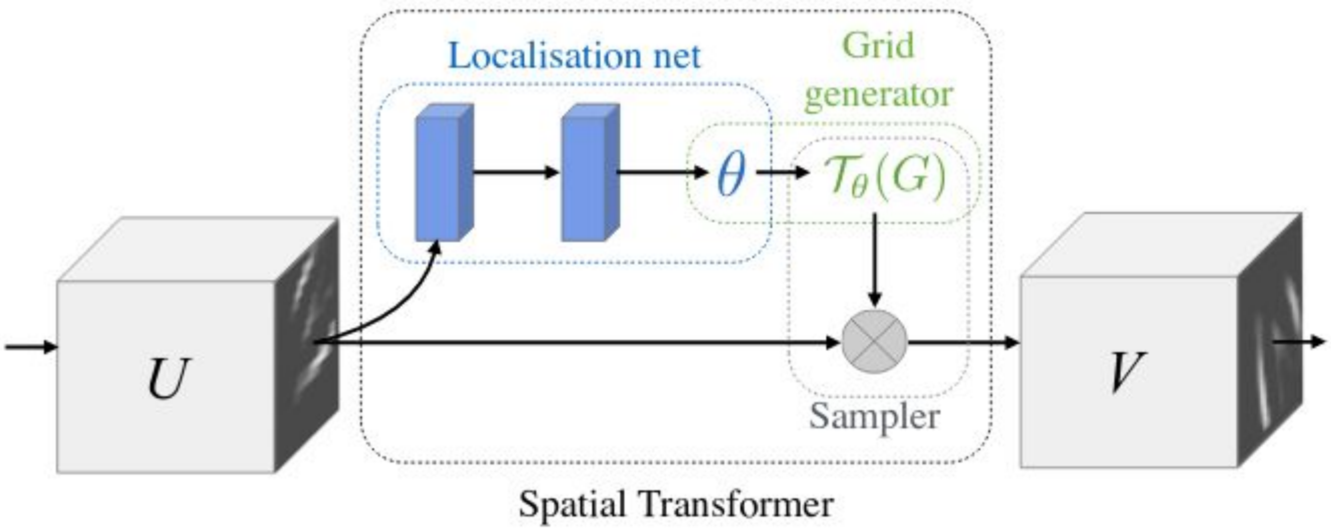
Architecture

Principle

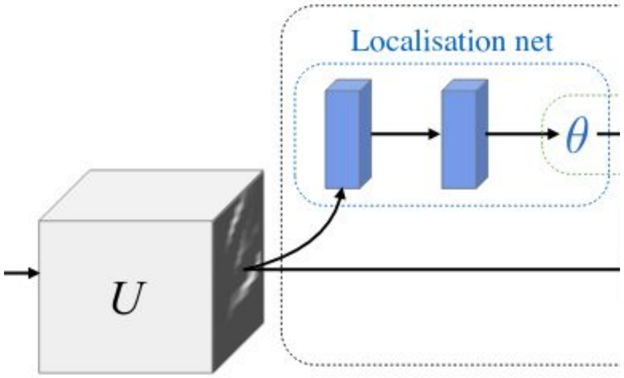
- Spatial transformer module
  - small
  - learnable (differentiable)
  - adjustable class of transformations
  - all channels

- Transformation
  - parameters - dependent on input image
  - grid generator
  - sampling

Architecture



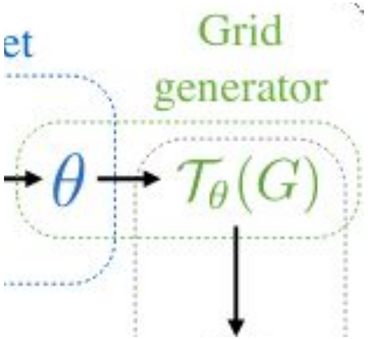
Localisation Network



- INPUT: $U \in \mathbb{R}^{H \times W \times C}$
- OUTPUT: θ parameters of transformation
- Localisation Network: $\theta = f_{loc}(U)$
- Localisation function could have any form FC, CNN… (But differentiable)

Sampling Grid

- Output pixels: $G = G_i,\ where\ G_i = (x_i^t, y_i^t)$
- Source coordinates: (x_i^s, y_i^s)



- Affine transformation example:

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

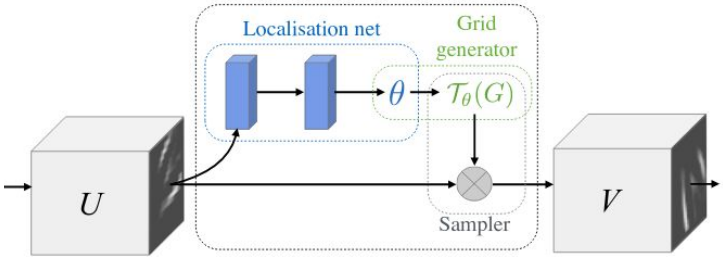
Transformations

- cropping, translation, isotropic scaling
- plane projective transformation
- differentiable with respect to the parameters

$$\mathtt{A}_\theta = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix}$$

Image Sampling

- input:
  - feature map $U \in \mathbb{R}^{H \times W \times C}$
  - sampling points $T_\theta(G) \to (x_i^s, y_i^s)$
- output:
  - feature map $V \in \mathbb{R}^{H' \times W' \times C}$
- Image sampling function :
  - differentiable
  - interpolation kernel $k()$
  - learnable kernel parameters Φ_x, Φ_y



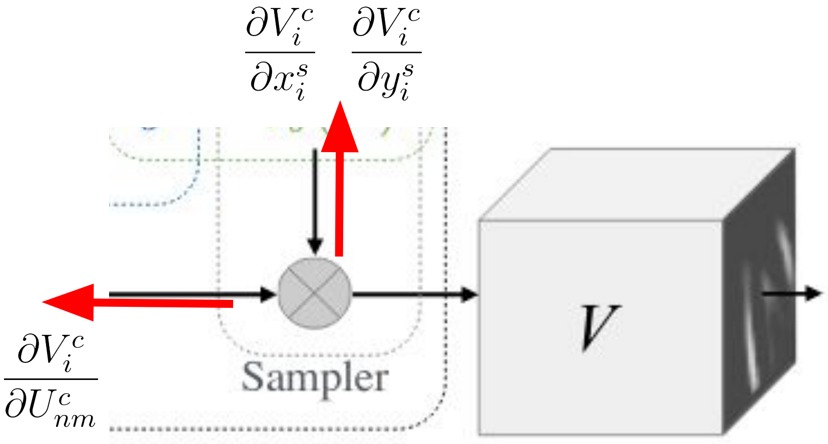
$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y) \ \forall i \in [1 \ldots H'W'] \ \forall c \in [1 \ldots C]$$

Image Sampling - example

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

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

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



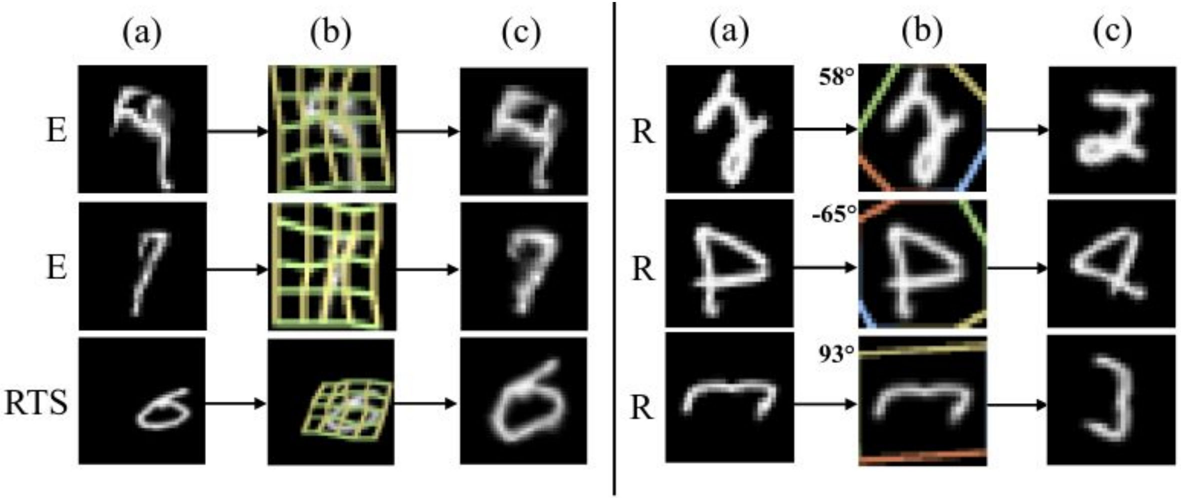
Spatial Transformer Networks

- self-contained module
- downsample, resample
- multiple or parallel

Experiments

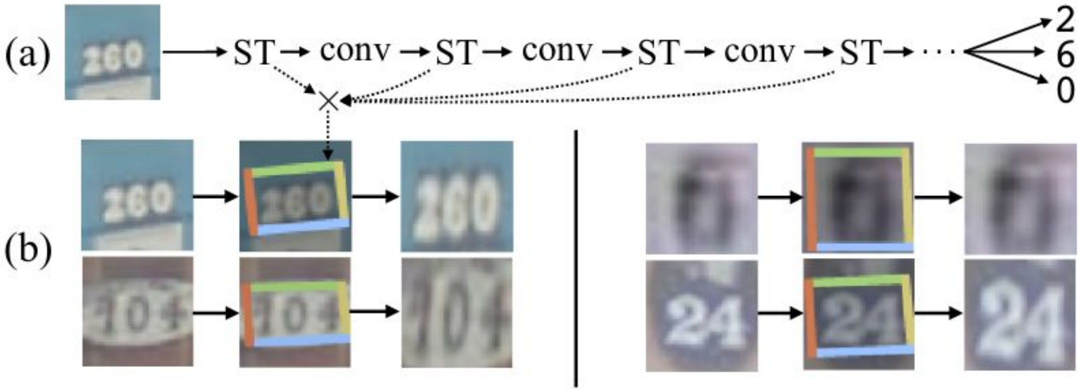
Distorted MNIST

Model		R	RTS	P	E
		MNIST Distortion			
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



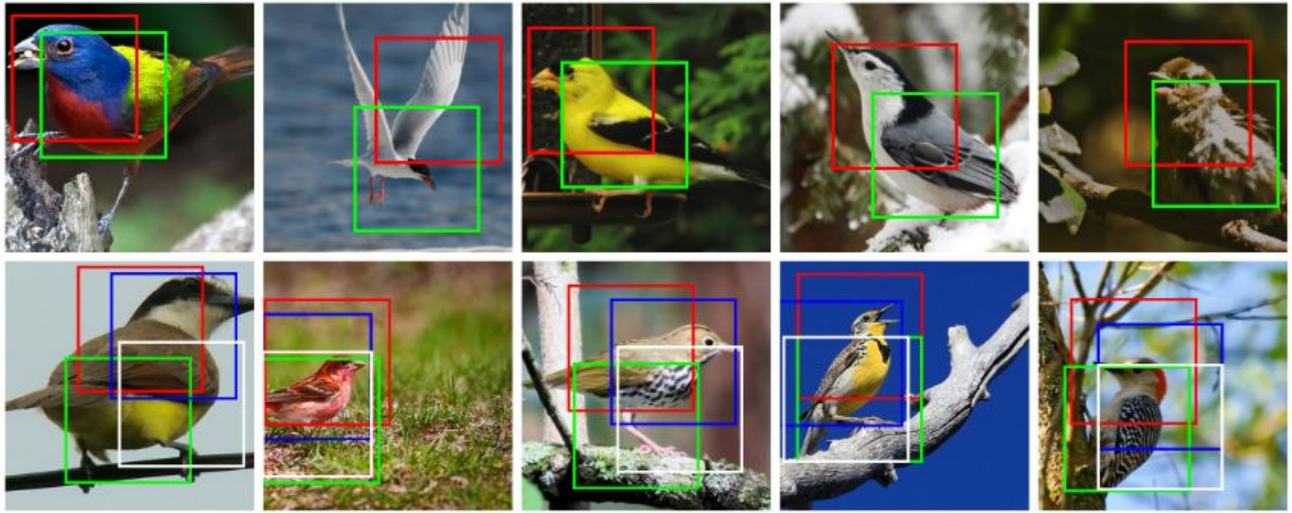
Street View House Numbers

Model		Size 64px	Size 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**



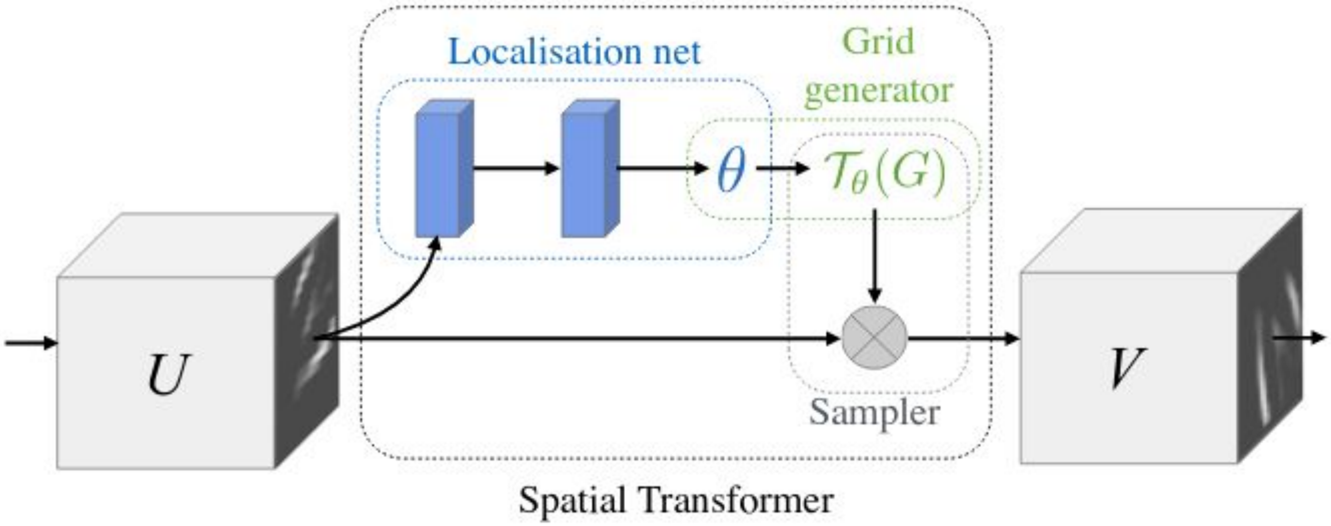
CUB-200-2011 - Classification

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**



Conclusion

Spatial Transformer




Thank you
for listening
© PNTS