

# Spatial transformer networks

Пузанова Анастасия  
НИУ ВШЭ

22 февраля 2019

# Содержание

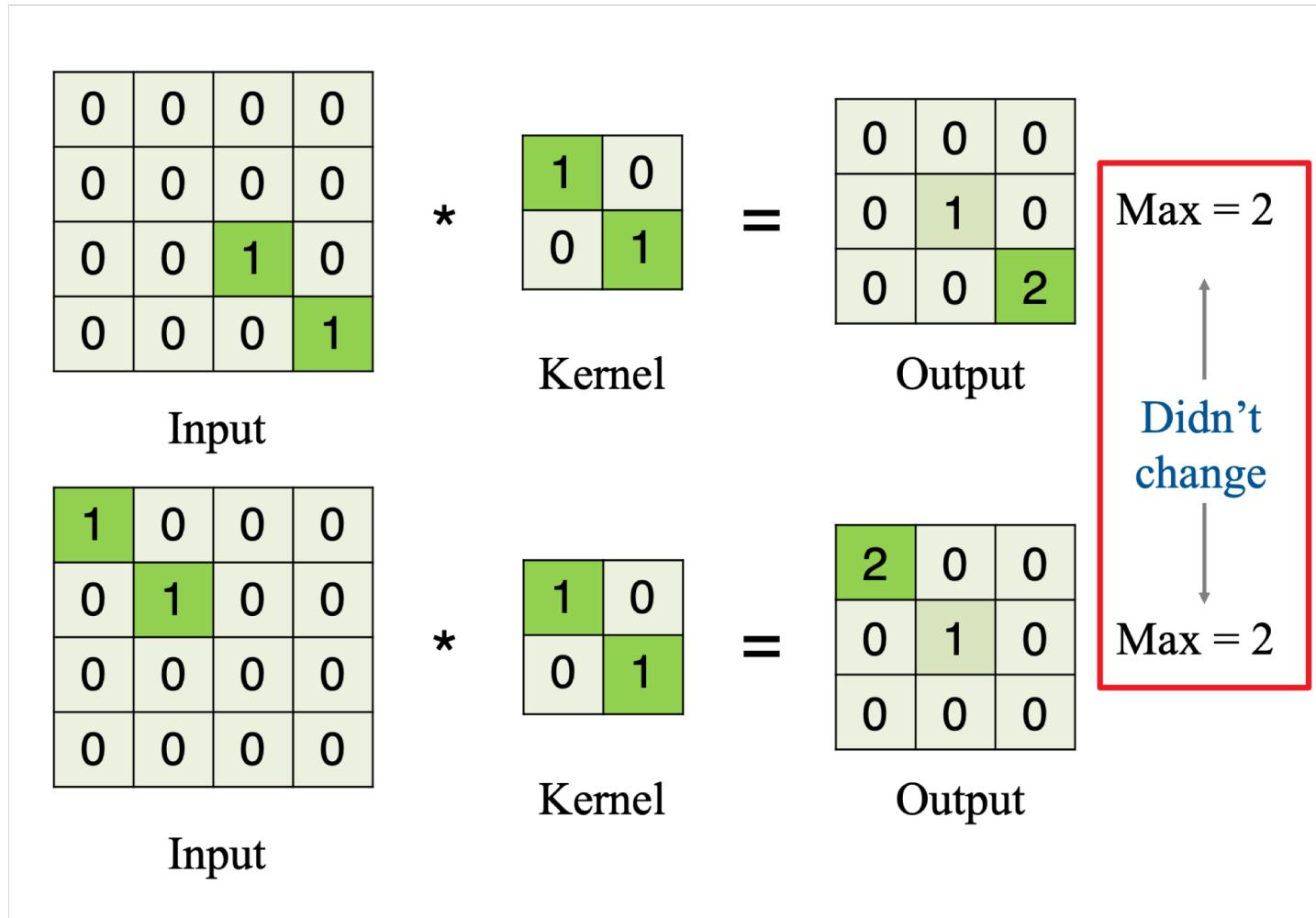
- Общая идея использования Spatial Transformer Networks (STN)
- Этапы преобразования:
  - Localization Network
  - Grid generator
  - Sampler
- Обзор применений:
  - Distorted MNIST
  - Fine-Grained Classification
- Выводы

# Постановка проблемы

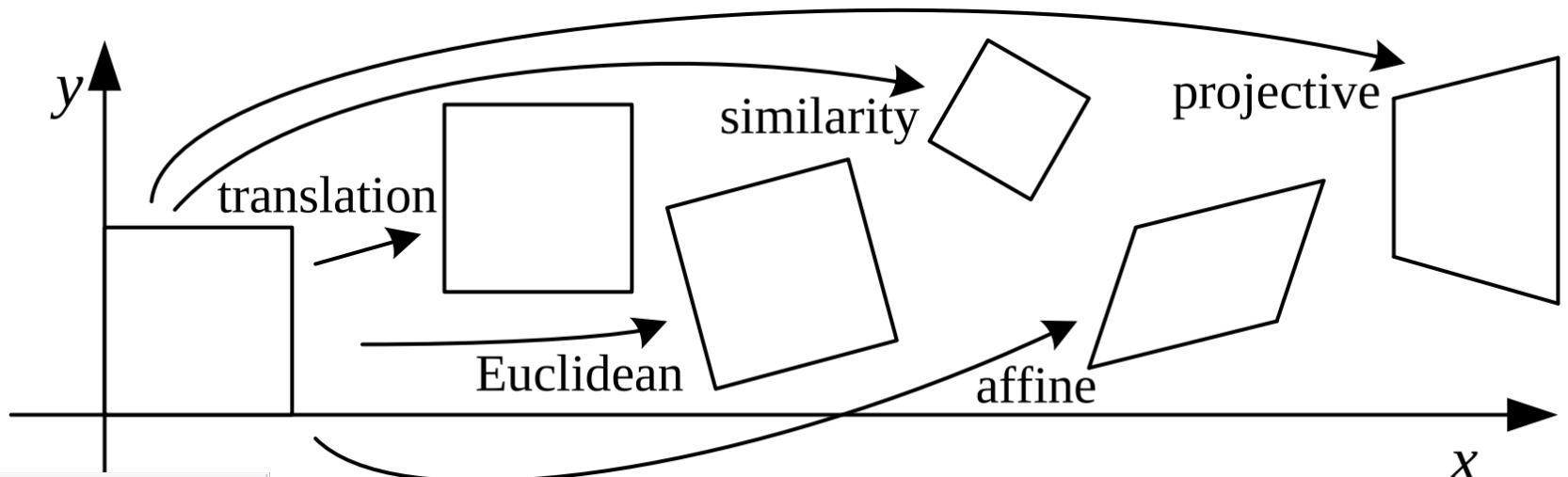
- Одна из проблем современных сверточных нейросетей – низкая инвариантность к входным данным
- Можно сказать, что max-pooling дает некоторую инвариантность, но в большинстве случаев ее недостаточно
- Можно делать аугментацию данных, но это не эффективно по памяти, требует значительных затрат по вычислительной мощности
- STN – lego-модуль, который позволяет решить проблему с пространственной инвариантностью CNN

# Инвариантность CNN+max-pooling

- Инвариантность к трансляции(параллельному переносу)



# Базовые геометрические преобразования

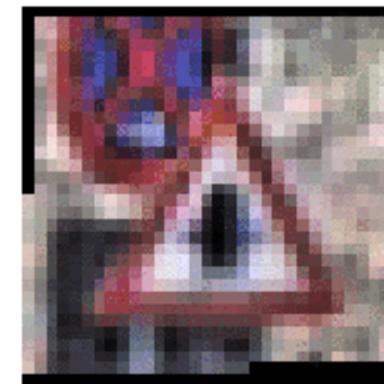


Transformation	Matrix Entries
Translation	$x' = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \bar{x}$
Rigid (Rotation + Translation)	$x' = \begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \end{bmatrix} \bar{x}$
similarity (scaled rotation)	$x' = \begin{bmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \end{bmatrix} \bar{x}$
Affine Transformation	$x' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \bar{x}$
Projection	$x' = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \bar{x}$

# Как работает STN?



batch = 0/200    theta =    0.98 0.02 -0.02  
    0.02 1.02 -0.02



## Краткое резюме

В общем, задача STN состоит в том, чтобы так геометрически преобразовать изображение/карту признаков, чтобы основная сеть-классификатор смогла проще определить нужный объект

## STN(Grid generator)

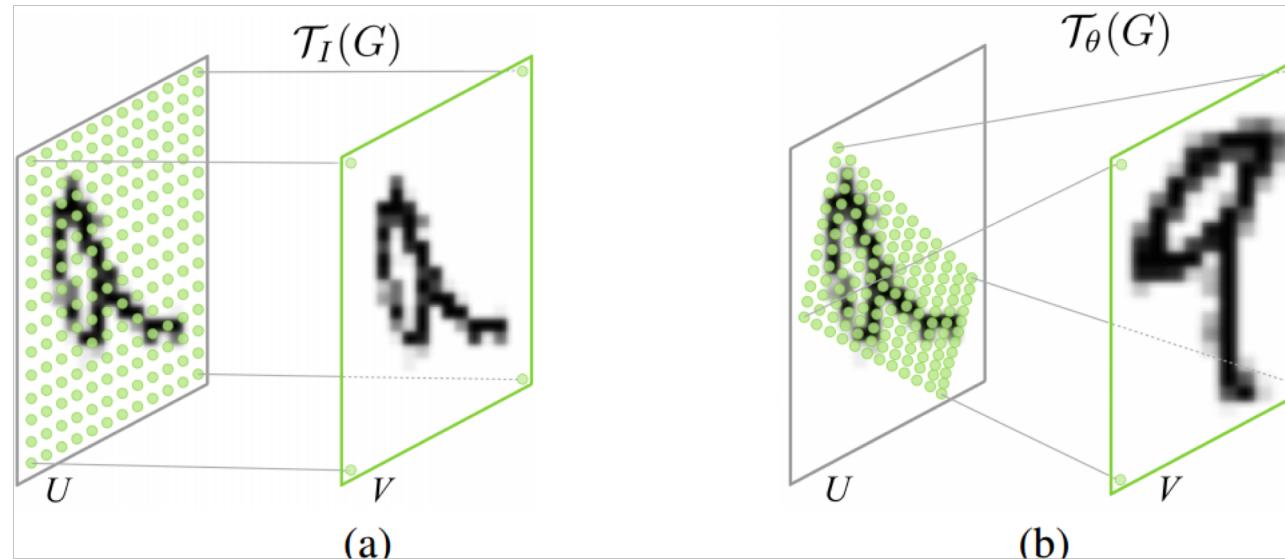
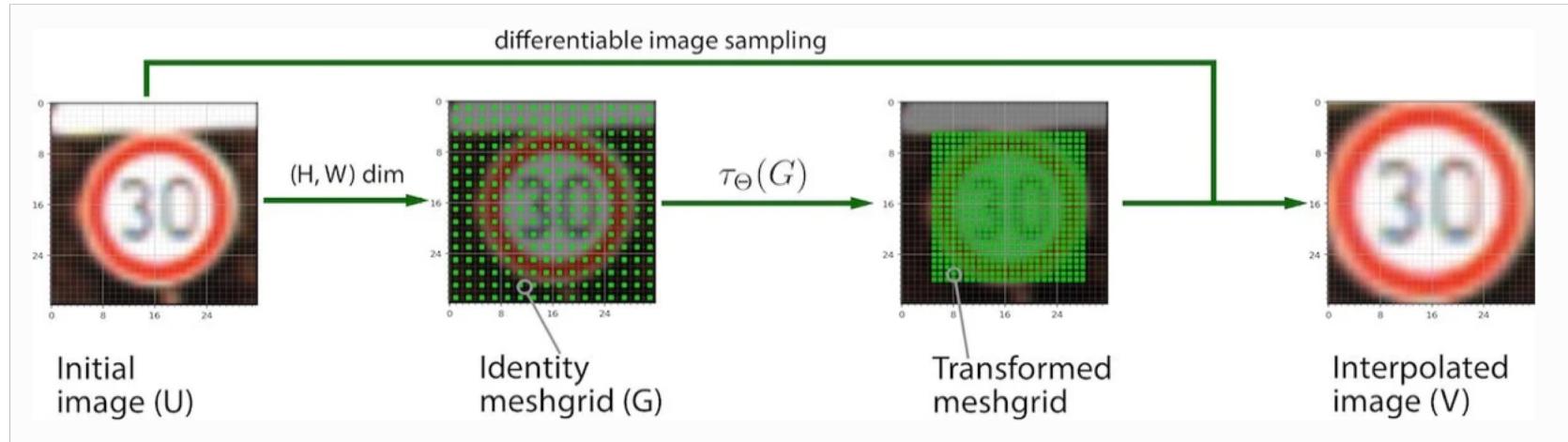
- Пусть нам известна матрица преобразования  $\Theta$
- Вместо того, чтобы преобразовывать исходное изображение, создадим равномерно покрывающую изображение выборочную сетку
- Применим матрицу преобразования к данной сетке. Получим новый набор точек.

$$\begin{bmatrix} x_i^s \\ y_i^s \end{bmatrix} = \Theta \begin{bmatrix} x_i^t \\ y_i^t \\ 1 \end{bmatrix}$$

, где  $x_i^s, y_i^s$  - новые точки,  $x_i^t, y_i^t$  - точки исходной сетки

- Наложим новую сетку на исходное изображение, получим точки измененного изображения

# STN(Grid generator)



# STN(Sampler)

- Данный этап состоит из двух шагов. Первый - применить интерполяцию, так как в результате преобразования выборочной сети координаты точек не обязательно будут целыми. При этом функция интерполяции должна быть дифференцируемой для корректной работы метода обратного распространения ошибки
- Второй - получить интерполированное изображение, используя исходную карту признаков и преобразованную выборочную сетку

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

Diagram illustrating the sampling process:

- x coordinate in  $\tau_\Theta(G)$** : A blue box labeled  $x_i^s$  representing the sampled coordinate.
- parameters of sampling kernel**: A blue box labeled  $k(\cdot; \Phi_x)$  representing the sampling kernel parameters.
- value at location  $(n, m)$  in channel  $c$  of input  $U$** : A blue box labeled  $U_{nm}^c$  representing the value from the input feature map.
- interpolation kernel**: A blue box labeled  $k(\cdot; \Phi_y)$  representing the interpolation kernel parameters.
- pixel in a channel  $c$** : A light gray box labeled  $V_i^c$  representing the final sampled pixel value.
- channels**: A light gray box labeled  $C$  representing the number of channels.

- При билинейной интерполяции соответственно:

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

# STN(Localization Network)

- Если бы матрица преобразования  $\Theta$  была известна, можно было бы, используя вышеописанное, сразу изменять изображение
- Мы хотим не просто подбирать  $\Theta$ , а получать ее с помощью машинного обучения из данных
- Для этого используется нейросеть, которая по изображению/карте признаков получает параметры преобразования, задающие матрицу  $\Theta$
- Данная нейросеть может быть практически любой(FCN, CNN), но ее последний слой не должен использовать нелинейную активацию
- Выход нейросети имеет фиксированный размер, который зависит от числа параметров преобразования

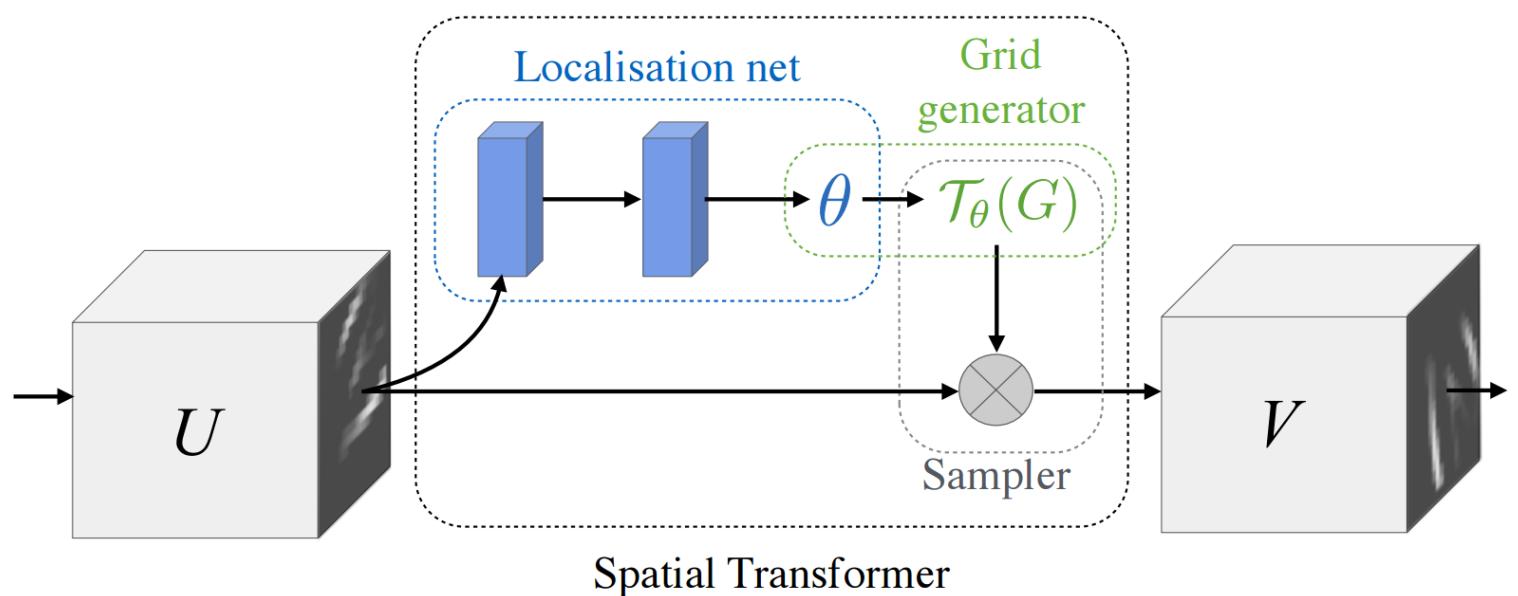
# Архитектура STN

Этапы преобразования:

- Localization Network
- Grid generator
- Sampler

U и V – входная и выходная карты признаков

Θ – параметры геометрического преобразования



# STN vs CNN

Варианты  $\Theta$ :

Aff – affine transformation

Proj – projective transformation

TPS – thin plate spline

Преобразования исходных данных:

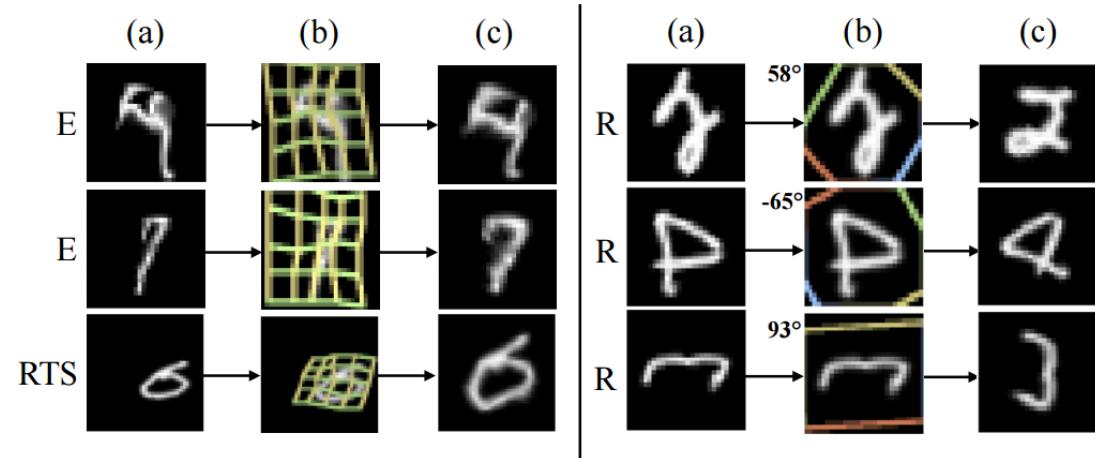
R – rotation

RTS – rotation+scaling+translation

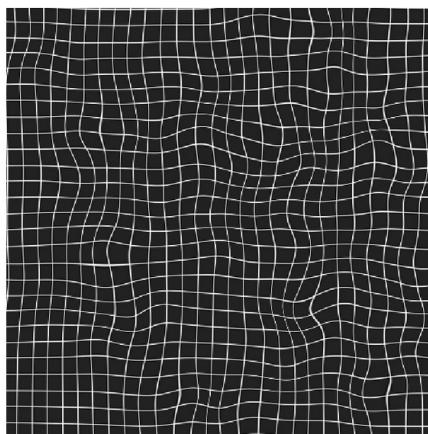
P – projective transformation

E – elastic warping

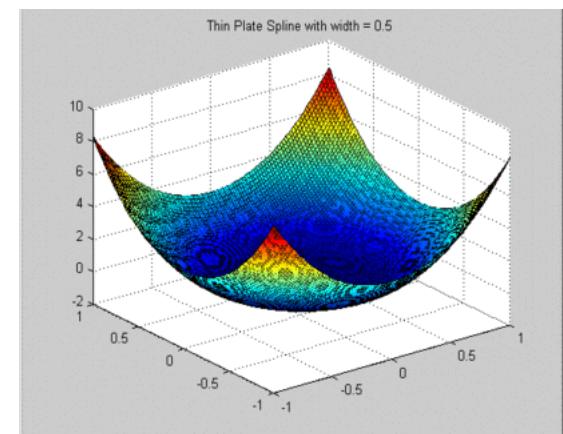
Model	MNIST Distortion			
	R	RTS	P	E
FCN	2.1	5.2	3.1	3.2
CNN	1.2	0.8	1.5	1.4
ST-FCN	Aff	1.2	0.8	1.5
	Proj	1.3	0.9	1.4
	TPS	1.1	0.8	2.4
ST-CNN	Aff	0.7	0.5	0.8
	Proj	0.8	0.6	0.8
	TPS	0.7	0.5	0.8



- MNIST dataset
- FCN и CNN vs STN+FCN и STN+CNN
- С использованием STN вне зависимости от  $\Theta$  или вида преобразования исходных данных ошибка уменьшается



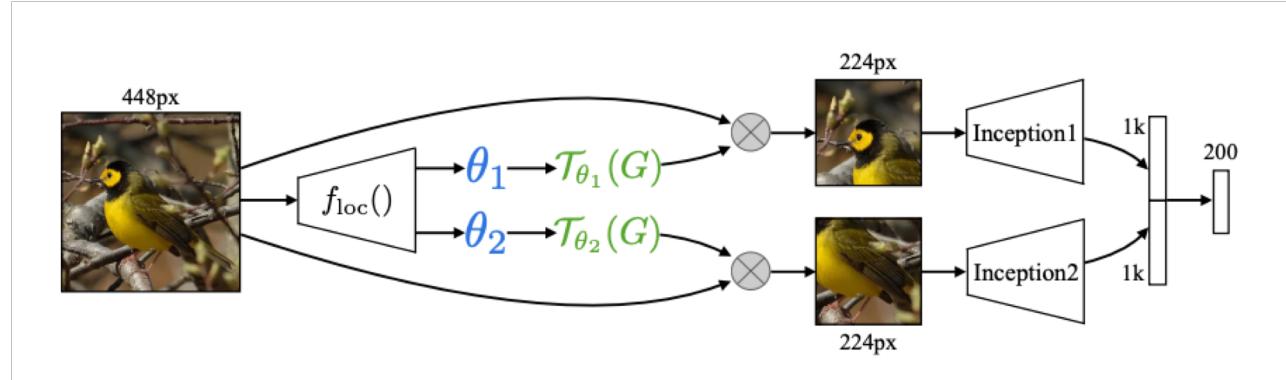
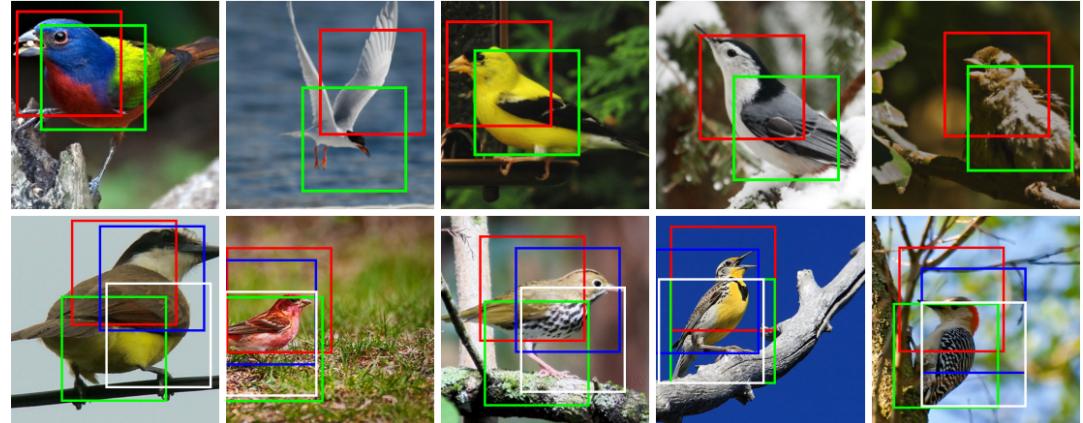
Elastic warping



TPS

# Sub-Object Classification

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	<b>84.1</b>



Зеленый - центральная часть туловища  
Красный - голова

- CUB-200-2011 bird dataset
- Изображения содержат разные виды птиц, сняты с разных ракурсов, на разном фоне
- Параллельные модули STN выбирают разные 'части' птицы для дальнейшего обучения

## Выводы

- STN — дифференцируемый модуль, который может быть интегрирован в сверточную нейронную сеть
- Блок STN работает по большей части самостоятельно, обучаясь на градиентах, приходящих от основной сети
- STN сэмплер применяет аффинное преобразование к исходным изображениям (или карте признаков) и получает трансформированное изображение/карту признаков
- Добавление одного или нескольких STN модулей в CNN усложняет обучение, делает его нестабильным: необходимо следить, чтобы обе сети не переобучались.

## Источники

- [1] – Köppl, Kilian. «Spatial Transformer Networks», 2017
- [2] - Кирилл Данилюк. «Распознавание дорожных знаков с помощью CNN: Spatial Transformer Networks»
- [3] – Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu. «Spatial Transformer Networks», 2016