2017-01-19
영상이해연구회 겨울학교

영상의 의미론적 정합을 위한 Deep Learning 기반 특징 추출

Dongbo Min

Department of Computer Science and Engineering

Chungnam National University, Korea

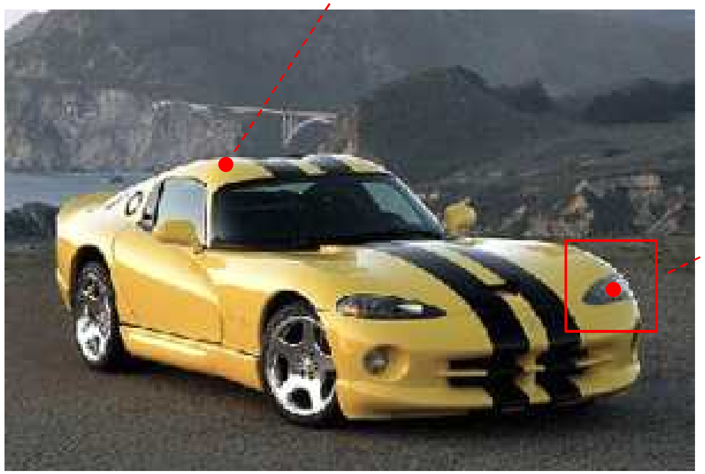
E-mail: dbmin@cnu.ac.kr **Web:** http://cvlab.cnu.ac.kr/

Acknowledgement: Dr. Jiangbo Lu (ADSC), Prof. Minh N. Do (UIUC), Seungryong Kim (Yonsei), Prof. Kwanghoon Sohn (Yonsei)

Local Image Descriptors

- **Objective**
  - Providing visual cues for establishing visual correspondence among multiple images

$(R_i, G_i, B_i)^T$: simple 3-D feature descriptor



$\bigcup_{j \in N_i} (R_i, G_i, B_i)^T$: simple 75-D feature descriptor (when using 5×5 window)

Such simple representations do NOT work well in many computer vision tasks.
Q: Is there a better way of doing this?

Image Descriptors

- Most features can be thought of as templates, histograms (counts), or combinations in hand-crafted descriptors

- The ideal descriptor should be
  - Robust
  - Distinctive
  - Compact
  - Efficient

- Most available descriptors focus on edge/gradient information
  - Capture texture information
  - Color rarely used

Slide courtesy from K. Grauman, B. Leibe

Image Descriptors

- **Hand-crafted descriptors**
  - SIFT, BRISK, BRIEF, Affine SIFT (ASIFT)
  - DAISY, Local Self-Similarity (LSS), Locally Adaptive Regression Kernels (LARK)
  - Rank Transform, Census transforms, Mutual Information (MI), Normalized Cross-Correlation (NCC), Zero-mean Normalized Cross-Correlation (ZNCC), Dense Adaptive Self-Correlation (DASC), Deep Self-Correlation (DSC) Descriptor
- **Learning-based descriptors**
  - *Brand new approaches* based on metric learning or convolutional neural networks (CNNs)

Local image descriptors and matching similarity measures

- **Descriptors for matching (sparse) interest points**
  - SIFT [1], BRISK [2], BRIEF [3], Affine SIFT (ASIFT) [4]

- **Descriptors for dense wide-baseline matching**
  - DAISY [5]

- **Descriptors for semi-dense large displacement matching**
  - Deep Matcher [6]

- **Descriptors for matching semantically similar image parts (e.g. cross-domain matching)**
  - Local Self-Similarity (LSS) [7], Locally Adaptive Regression Kernels (LARK) [8]

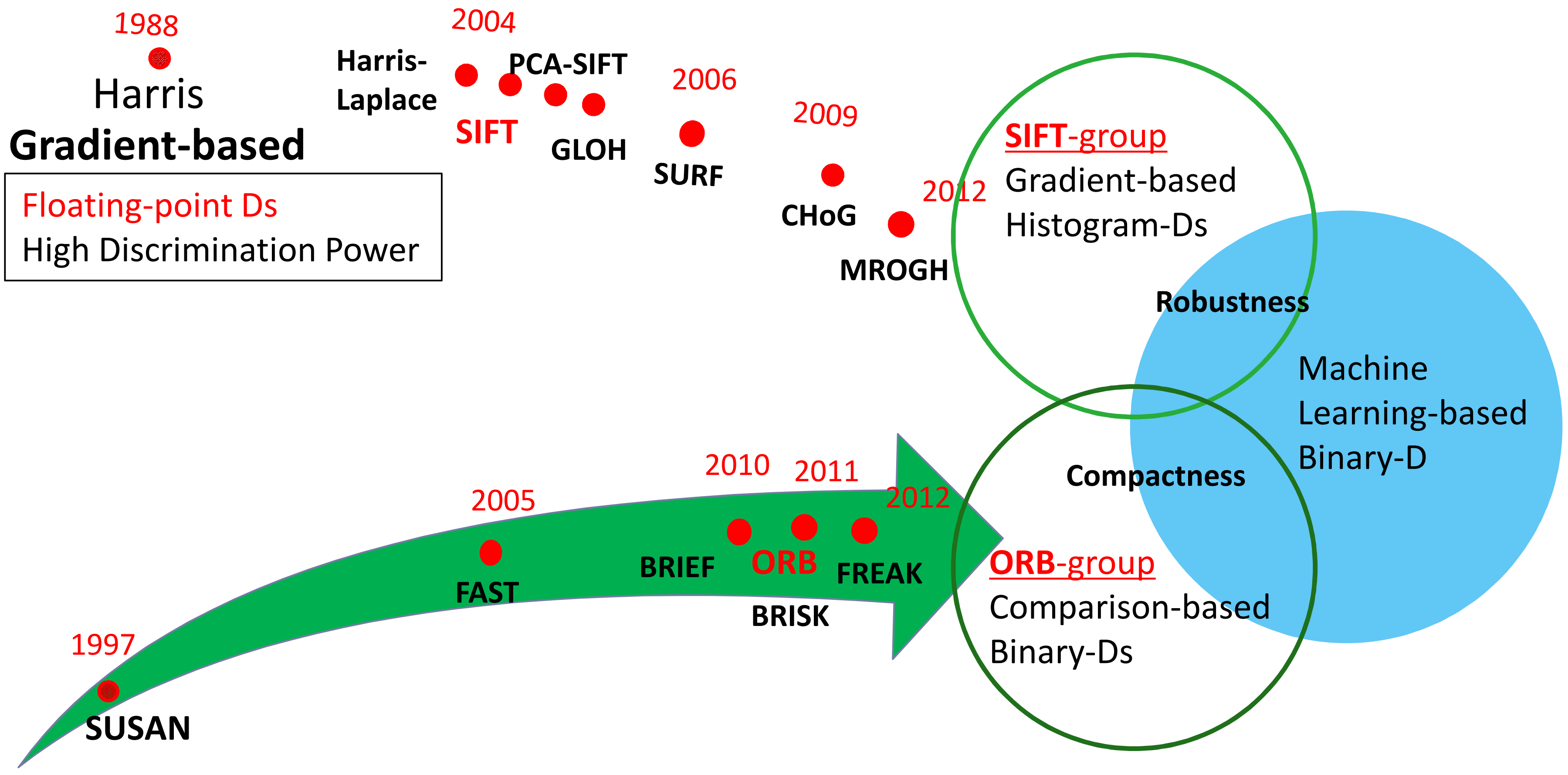
- **Similarity measures for handling photometric and multi-modal variations**
  - Rank Transform, Census transforms [9], Mutual Information (MI) [10], Normalized Cross-Correlation (NCC) [11], Zero-mean Normalized Cross-Correlation (ZNCC) [12], Dense Adaptive Self-Correlation (DASC) [13,14], Deep Self-Correlation (DSC) Descriptor [15]

- **Learning based descriptors**
  - Measure the patch similarity using CNNs [18], [19], [20], [21]

Reference - Descriptor

1. D. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. Journal of Computer Vision, 2004.
2. S. Leutenegger, et al., “BRISK: Binary robust invariant scalable keypoints,” ICCV 2011.
3. M. Calonder, et al., “BRIEF: Computing a local binary descriptor very fast,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 2012.
4. J. M. Morel and G. Yu, “ASIFT: A new framework for fully affine invariant image comparison,” SIAM Journal on Imaging Sciences, 2009.
5. E. Tola, V. Lepetit, and P. Fua, “DAISY: An efficient dense descriptor applied to wide-baseline stereo,” IEEE Trans. Pattern Analysis and Machine Intelligence, 2010.
6. P. Weinzaepfel, J. Revaud, Z Harchaoui, and C. Schmid, “DeepFlow: Large displacement optical flow with deep matching,” ICCV 2013.
7. E. Schechtman and M. Irani, “Matching local self-similarities across images and videos,” CVPR 2007.
8. H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” IEEE Trans. on Image Processing, 2007.
9. R. Zabih and J. Woodfill, “Non-parametric local transforms for computing visual correspondence,” ECCV 1994.
10. H. Hirschmuller, “Stereo processing by semi-global matching and mutual information,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 2008.
11. Y. S. Heo, K. M. Lee, and S. U. Lee, “Robust stereo matching using adaptive normalized cross-correlation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 2011.
12. X. Shen, L. Xu, Q. Zhang, and J. Jia, “Multi-modal and multi-spectral registration for natural images,” ECCV 2014.
13. S. Kim, D. Min, B. Ham, S. Ryu, M. N. Do, and K. Sohn, “DASC: Dense Adaptive Self-Correlation Descriptor for Multi-modal and Multi-spectral Correspondence,” CVPR 2015.
14. S. Kim, D. Min, B. Ham, M. N. Do, and K. Sohn, “DASC: Robust Dense Descriptor for Multi-modal and Multi-spectral Correspondence Estimation,” IEEE Trans. on Pattern Analysis and Machine Intelligence. (In press)
15. S. Kim, D. Min, S. Lin, and K. Sohn, “Deep Self-Correlation Descriptor for Dense Cross-Modal Correspondence,” ECCV 2016
16. H. Hirschmuller and D. Scharstein, “Evaluation of stereo matching costs on images with radiometric differences,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 2009.
17. C. Vogel, S. Roth, and K. Schindler, “An Evaluation of Data Costs for Optical Flow,” GCPR 2013.
18. J. Zbontar and Y. LeCun, “Computing the Stereo Matching Cost With a Convolutional Neural Network,” CVPR, 2015
19. S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” CVPR, 2015
20. E. Simo-Serra, et al, “Discriminative learning of deep convolutional feature point descriptors,” ICCV, 2015
21. C. B. Choy, Y. Gwak, and S. Savarese, “Universal Correspondence Network,” NIPS, 2016

Detectors and Descriptors for Hand-Crafted Features



1988

Harris

Gradient-based

Floating-point Ds
High Discrimination Power

2004

Harris-Laplace

PCA-SIFT

SIFT

GLOH

2006

SURF

2009

CHoG

2012

MROGH

SIFT-group
Gradient-based
Histogram-Ds

Robustness

Machine
Learning-based
Binary-D

Compactness

1997

SUSAN

2005

FAST

2010

BRIEF

2011

ORB

2012

FREAK

BRISK

ORB-group
Comparison-based
Binary-Ds

Intensity-based
Wedge model detector

Binary Ds
Less Storage Fast Matching

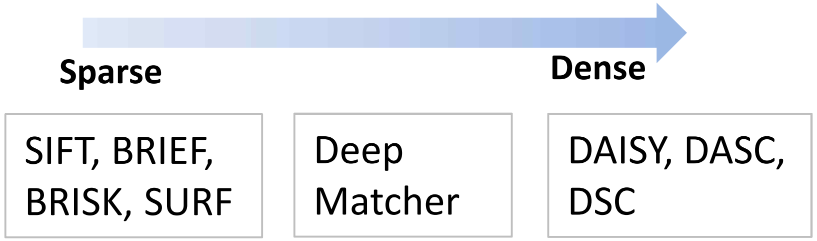
“ORB: an efficient alternative to SIFT or SURF”

Courtesy from Prof. Chee Sun Won

Challenges of Image Descriptors

Note that here we show the hand-crafted descriptors, as the performance of learning based descriptors are not fully studied yet!

Density (Considering computational redundancy!)



Sparse

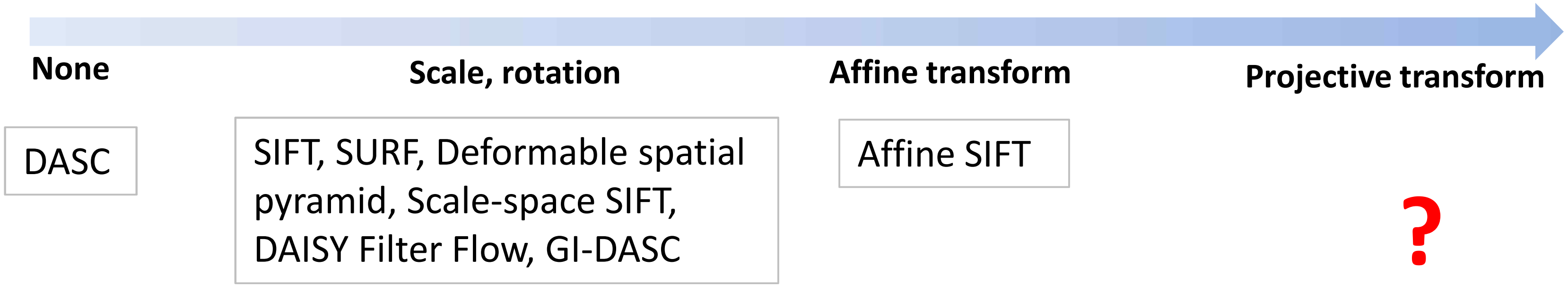
Dense

SIFT, BRIEF, BRISK, SURF

Deep Matcher

DAISY, DASC, DSC

Geometric Distortion



None

Scale, rotation

Affine transform

Projective transform

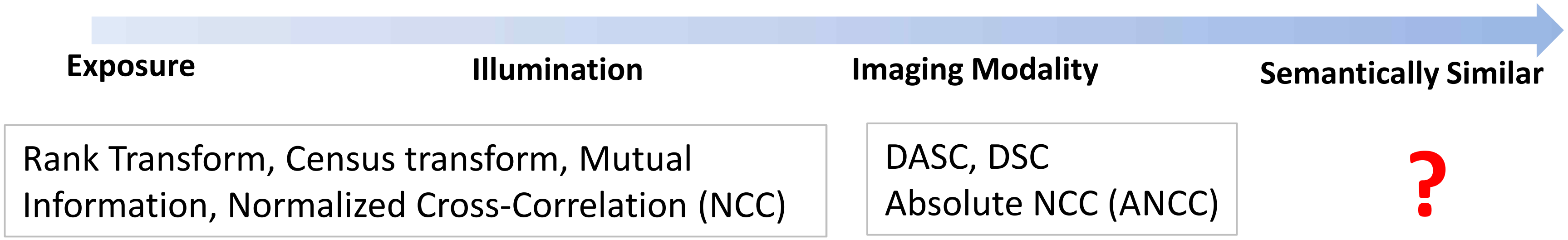
DASC

SIFT, SURF, Deformable spatial pyramid, Scale-space SIFT, DAISY Filter Flow, GI-DASC

Affine SIFT

?

Photometric Distortion



Exposure

Illumination

Imaging Modality

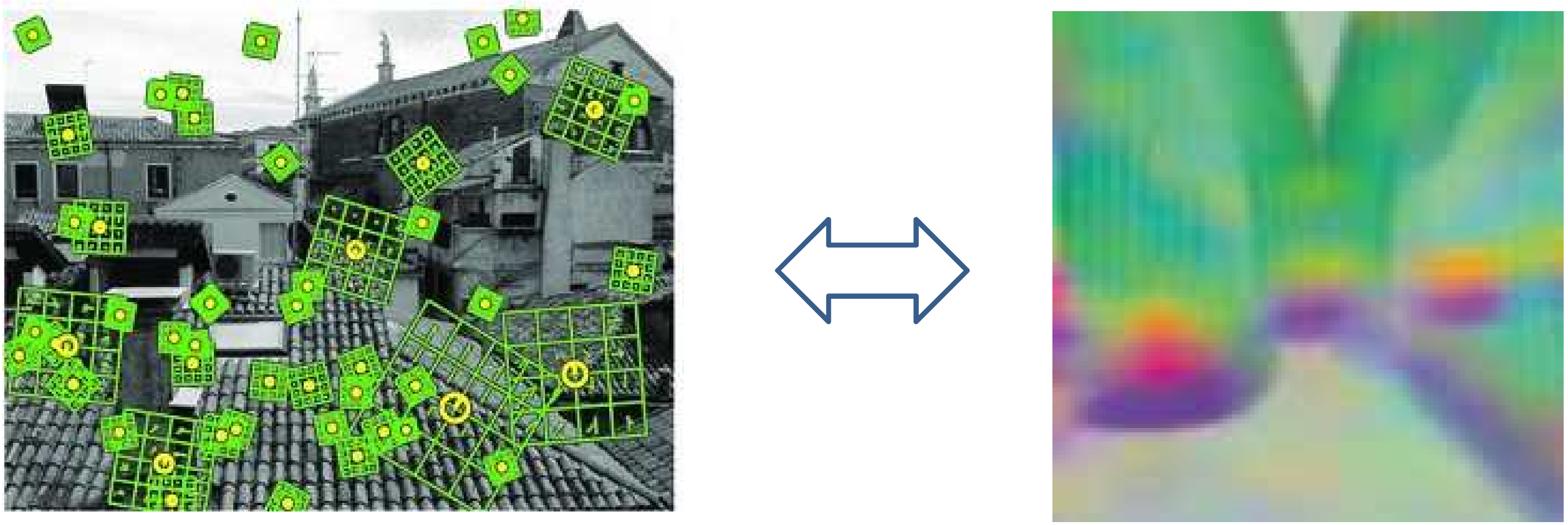
Semantically Similar

Rank Transform, Census transform, Mutual Information, Normalized Cross-Correlation (NCC)

DASC, DSC
Absolute NCC (ANCC)

?

Density



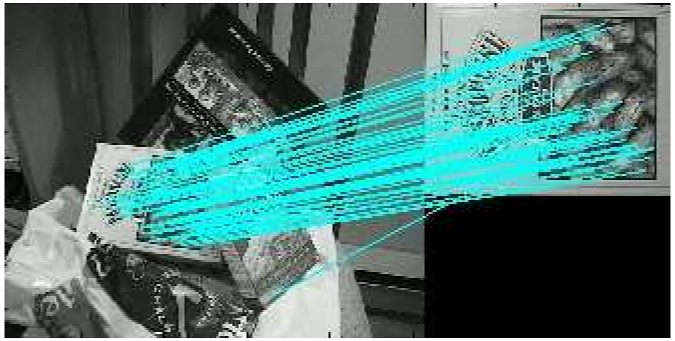
The density of descriptors depends on applications

- **Sparse feature**: Camera tracking, image retrieval, Structure-from-Motion
- **Dense feature**: annotation propagation, dense semantic labeling, depth or motion recovery

Densely computing the descriptors provokes *a huge amount of computational complexity*!

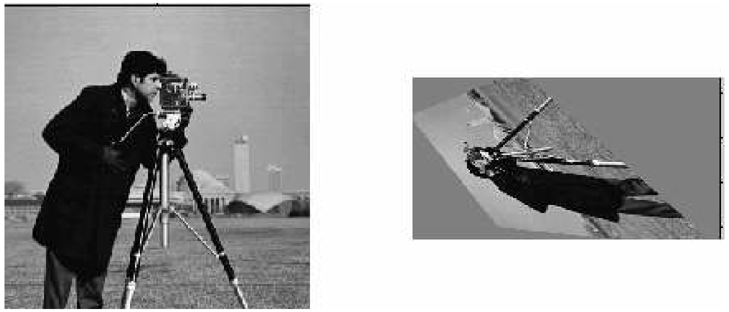
Geometric Distortion

Scale and rotation variations



Affine transform variation

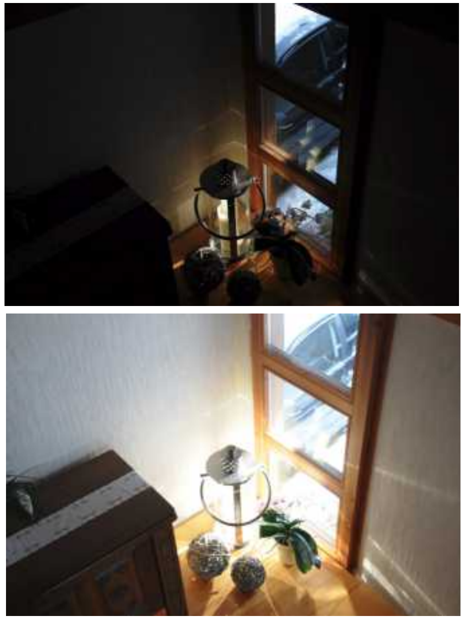
Projective transform variation



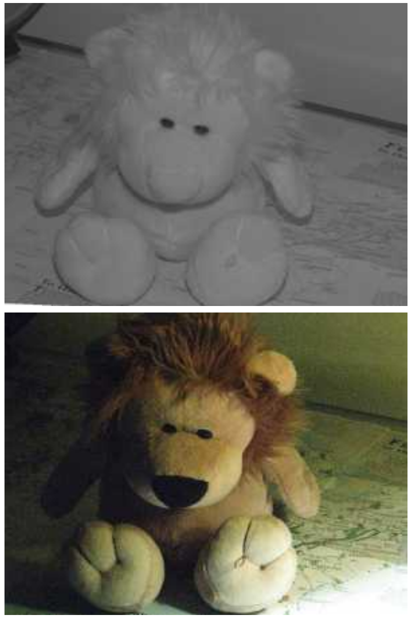
Even state-of-the-arts hand-crafted descriptors can mostly handle scale and rotation variations.

Photometric Distortion

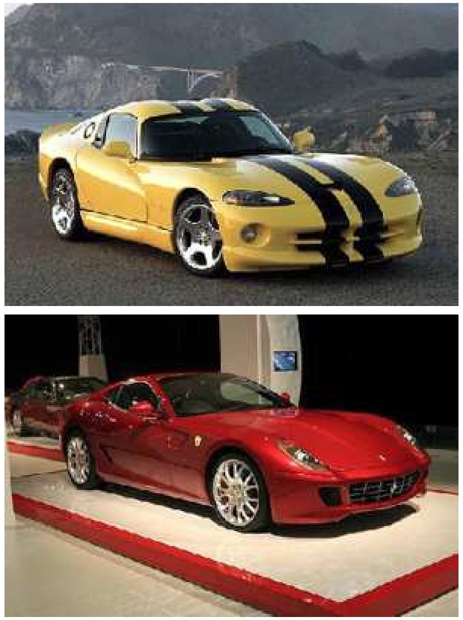
Exposure or illumination



Imaging Modality



Intra class variation:
semantically similar objects



Even state-of-the-arts hand-crafted descriptors can handle rather simple photometric distortions such as exposure or illumination variations to some extent.

Hand-crafted vs. Learning-based Descriptors

- **Hand-crafted descriptors**
  - "Distinctive image features from scale-invariant keypoints," Int. Journal of Computer Vision, 2004. (Sparse, scale/rotation/illumination invariant feature)
  - "DAISY: An efficient dense descriptor applied to wide-baseline stereo," IEEE Trans. Pattern Analysis and Machine Intelligence, 2010. (Dense, illumination invariant feature)
  - "DASC: Robust Dense Descriptor for Multi-modal and Multi-spectral Correspondence Estimation," IEEE Trans. on Pattern Analysis and Machine Intelligence, 2016 (In press) (Dense, scale/rotation/illumination invariant feature)

- **Learning-based descriptors**
  - "Computing the Stereo Matching Cost With a Convolutional Neural Network," CVPR, 2015 (Dense, illumination invariant feature)
  - "Learning to compare image patches via convolutional neural networks," CVPR, 2015 (Sparse, illumination invariant feature)
  - "Discriminative learning of deep convolutional feature point descriptors," ICCV, 2015 (Sparse, illumination invariant feature)
  - "Universal Correspondence Network," NIPS, 2016 (Dense, scale/rotation/illumination invariant feature)

Note that 'sparse' or 'dense' descriptors are classified, depending on whether the descriptor can be densely computed in an efficient manner.

PART 1.1: LEARNING-BASED DESCRIPTORS

Discriminative Learning of Descriptors using CNNs

J. Zbontar and Y. LeCun, “Computing the Stereo Matching Cost With a Convolutional Neural Network,” CVPR, 2015
S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” CVPR, 2015
E. Simo-Serra, et al, “Discriminative learning of deep convolutional feature point descriptors,” ICCV, 2015

Fully convolutional networks for semantic segmentation

- Received CVPR 2015 best paper award!
  - First work employing fully convolutional network for pixel-level labeling tasks

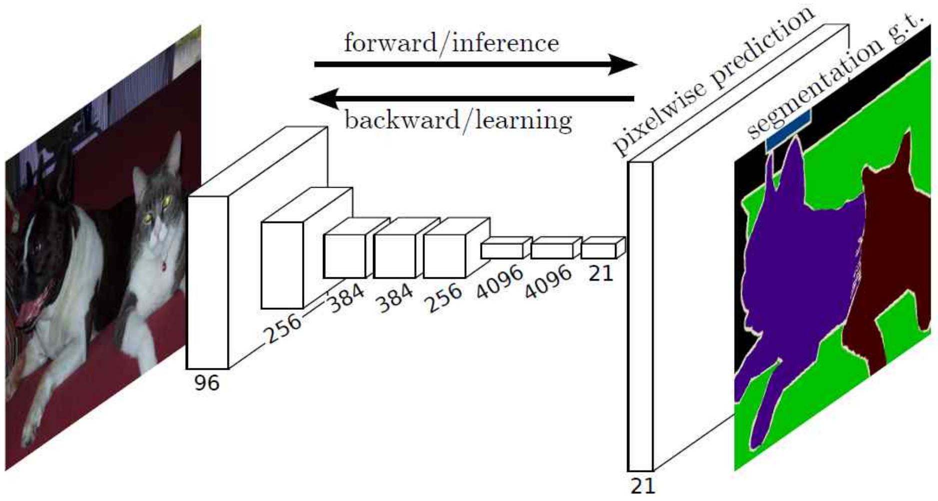
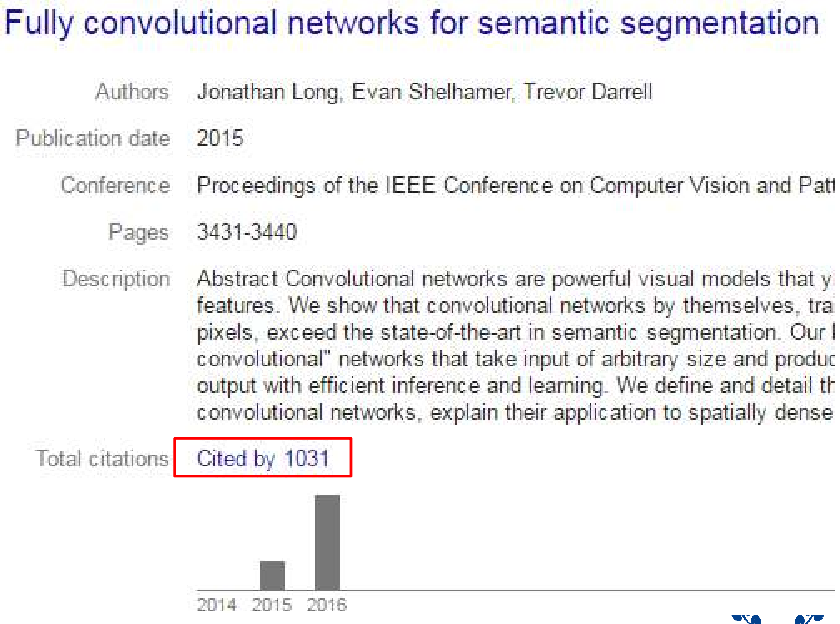


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.



Fully convolutional networks for semantic segmentation

Authors Jonathan Long, Evan Shelhamer, Trevor Darrell

Publication date 2015

Conference Proceedings of the IEEE Conference on Computer Vision and Patt

Pages 3431-3440

Description Abstract Convolutional networks are powerful visual models that y features. We show that convolutional networks by themselves, tra pixels, exceed the state-of-the-art in semantic segmentation. Our l convolutional" networks that take input of arbitrary size and produc output with efficient inference and learning. We define and detail th convolutional networks, explain their application to spatially dense

Total citations Cited by 1031

Fully convolutional networks for semantic segmentation

- But, this work relies on network architecture for a single task (semantic segmentation)

  Q: What if we wish to do different task? New architecture is needed

- This does NOT provide the location of components?

  Ex) Where is the ear of dog?

Next step

General purpose pixel-level image descriptors based on CNN are STRONGLY needed

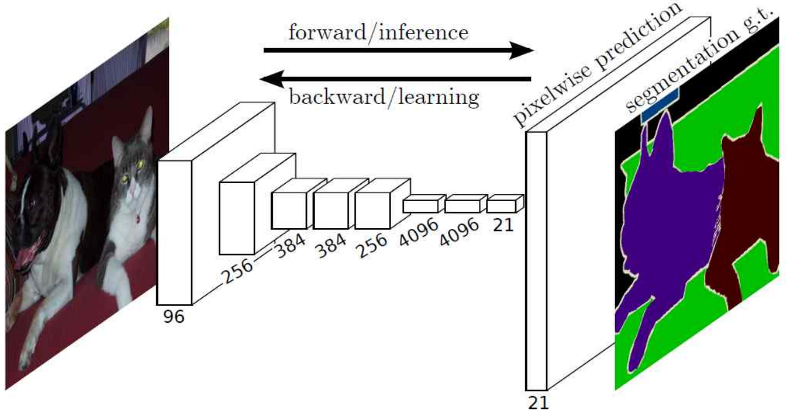
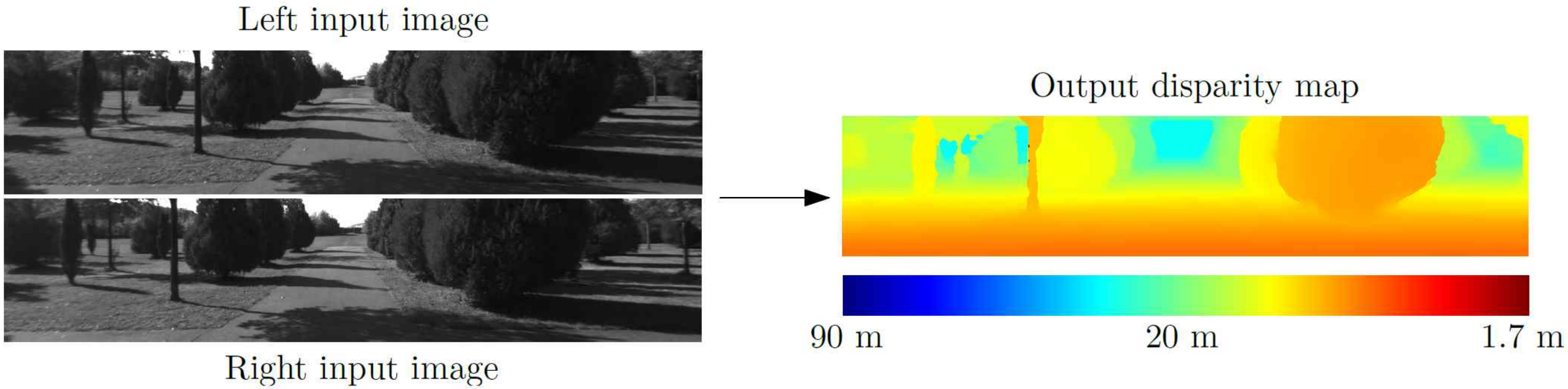


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

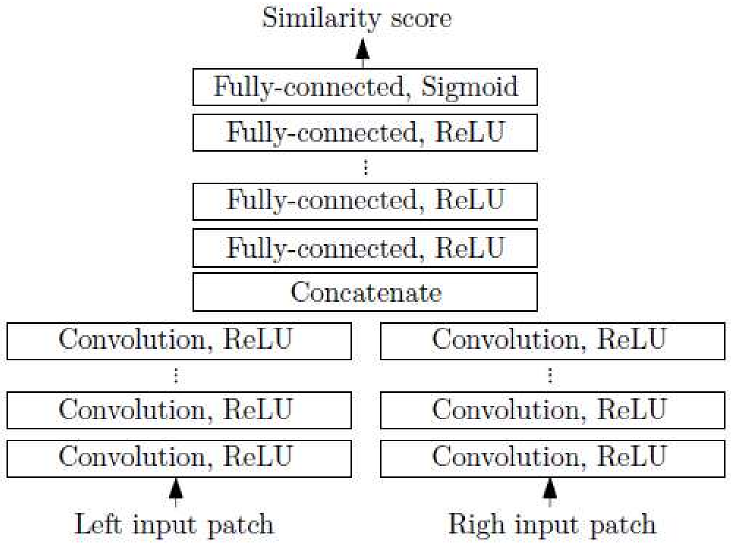
Matching Cost in Convolutional Neural Networks (MC-CNN)

- **Apply CNN to stereo matching!**

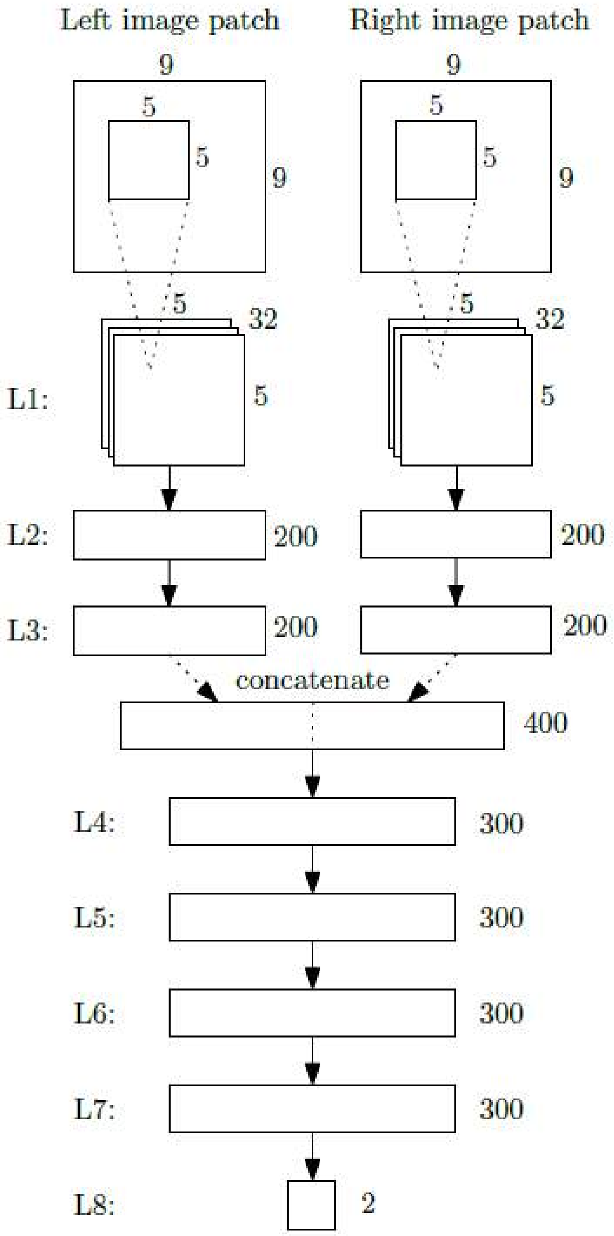


MC-CNN

- **Procedures**
  1. Train two patches (positive or negative samples)
  2. Measure a similarity value between two patches in test phase



=



MC-CNN

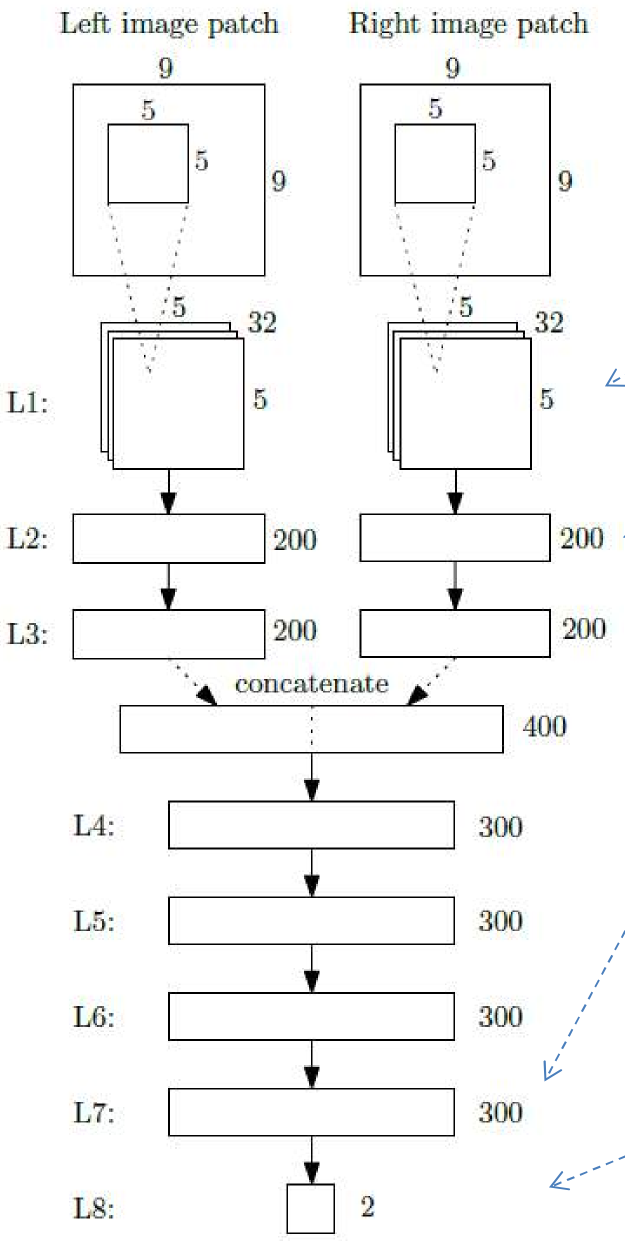
- Prepare training patches for positive and negative examples

$$< \mathcal{P}^{L}_{9\times9}(\mathbf{p}), \mathcal{P}^{R}_{9\times9}(\mathbf{q}) >$$

- Negative examples $\mathbf{q} = (x - d + o_{\text{neg}}, y)$
  - O_{neg} : an offset corrupting the match, chosen randomly from the set $\{-N_{\text{hi}}, \dots, -N_{\text{lo}}, N_{\text{lo}}, \dots, N_{\text{hi}}\}$

- Positive examples $\mathbf{q} = (x - d + o_{\text{pos}}, y)$
  - O_{pos} : chosen randomly from the set $\{-P_{\text{hi}}, \dots, P_{\text{hi}}\}$

Network Architecture



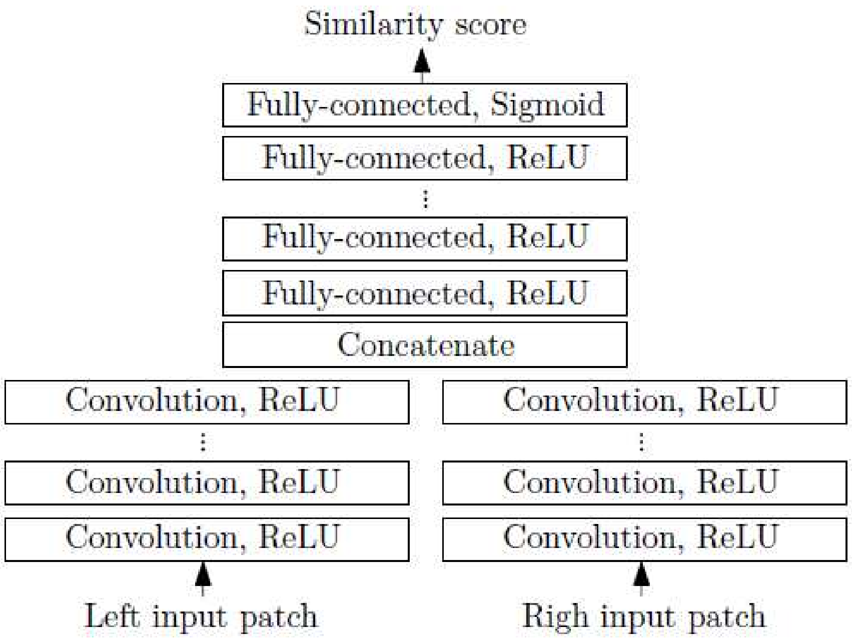
$5 \times 5 \times 32$
convolutional kernel

Fully connected layer

Note) L1, L2, and L3 of the networks for left and right patches are tied

Output with two real numbers that are fed through a softmax function, producing a distribution over the two classes (good match and bad match)

Fast Implementation of MC-CNN



Similarity score

Fully-connected, Sigmoid

Fully-connected, ReLU

⋮

Fully-connected, ReLU

Fully-connected, ReLU

Concatenate

Convolution, ReLU | Convolution, ReLU

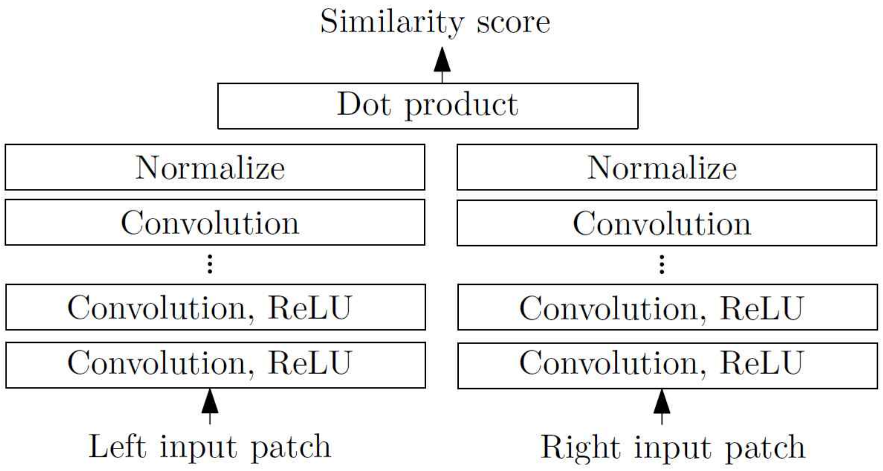
⋮ | ⋮

Convolution, ReLU | Convolution, ReLU

Convolution, ReLU | Convolution, ReLU

Left input patch | Righ input patch

Original accurate version



Similarity score

Dot product

Normalize | Normalize

Convolution | Convolution

⋮ | ⋮

Convolution, ReLU | Convolution, ReLU

Convolution, ReLU | Convolution, ReLU

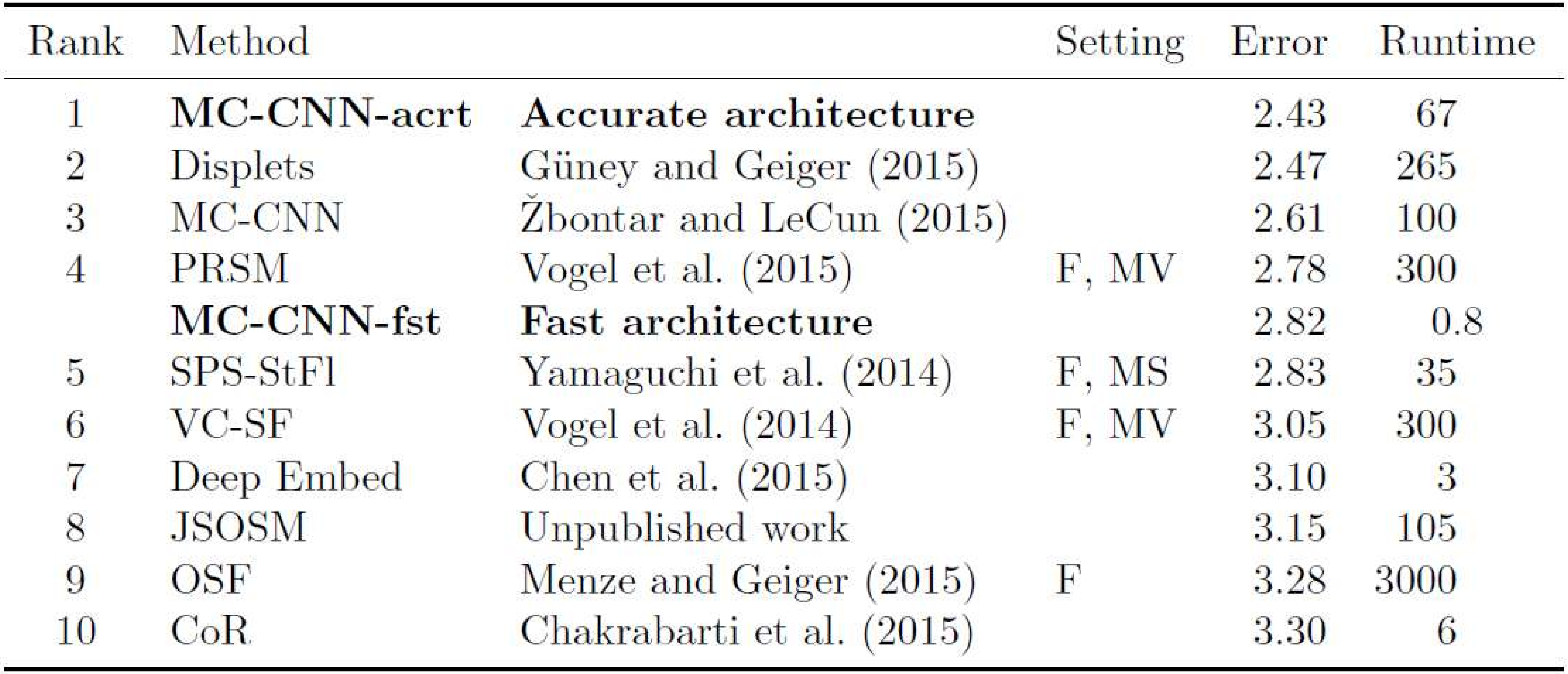
Left input patch | Right input patch

Fast version

J. Zbontar and Y. LeCun, “ Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches, Journal of Machine Learning Research, 2016 (Extension of CVPR 2015)

Outstanding Performance on Benchmark

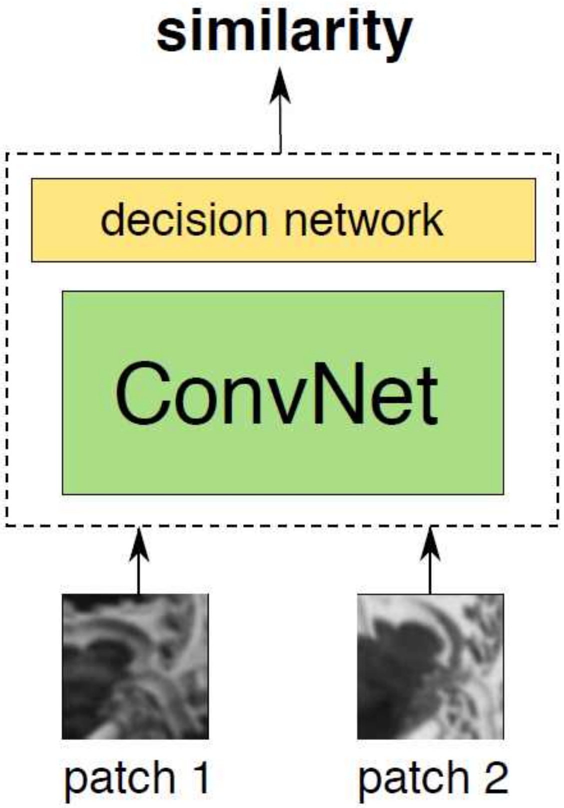
The highest ranking methods on the KITTI 2012 data set as of October 2015
Note) This simple CNN based method outperforms all state-of-the-arts approaches.



Rank	Method		Setting	Error	Runtime
1	**MC-CNN-acrt**	**Accurate architecture**		2.43	67
2	Displets	Güney and Geiger (2015)		2.47	265
3	MC-CNN	Žbontar and LeCun (2015)		2.61	100
4	PRSM	Vogel et al. (2015)	F, MV	2.78	300
	MC-CNN-fst	**Fast architecture**		2.82	0.8
5	SPS-StFl	Yamaguchi et al. (2014)	F, MS	2.83	35
6	VC-SF	Vogel et al. (2014)	F, MV	3.05	300
7	Deep Embed	Chen et al. (2015)		3.10	3
8	JSOSM	Unpublished work		3.15	105
9	OSF	Menze and Geiger (2015)	F	3.28	3000
10	CoR	Chakrabarti et al. (2015)		3.30	6

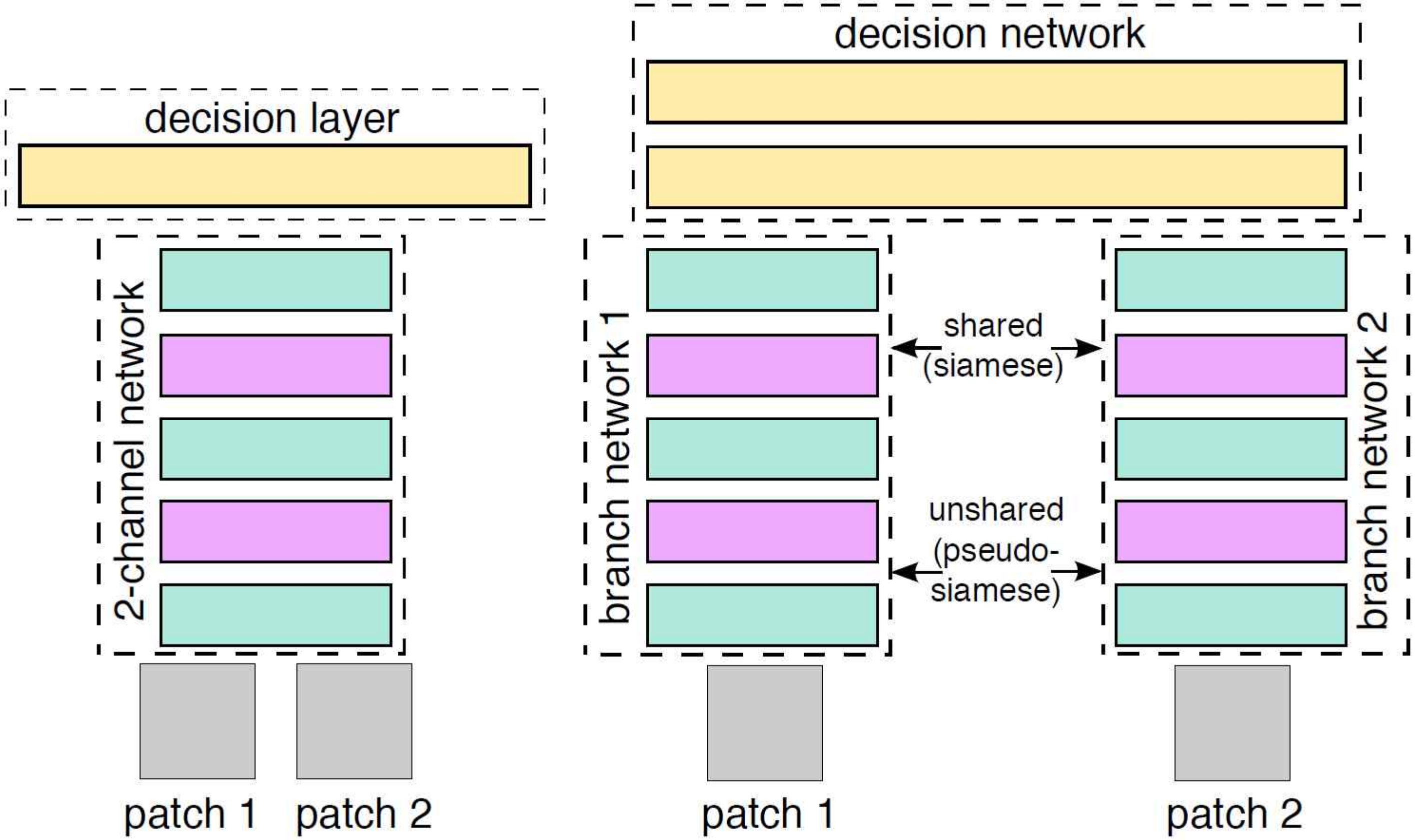
Learning to Compare Image Patches via CNNs, CVPR 2015

- **Goal**: learning a general similarity function for image patches
- Almost similar to MC-CNN, the method models the patch similarity using CNNs



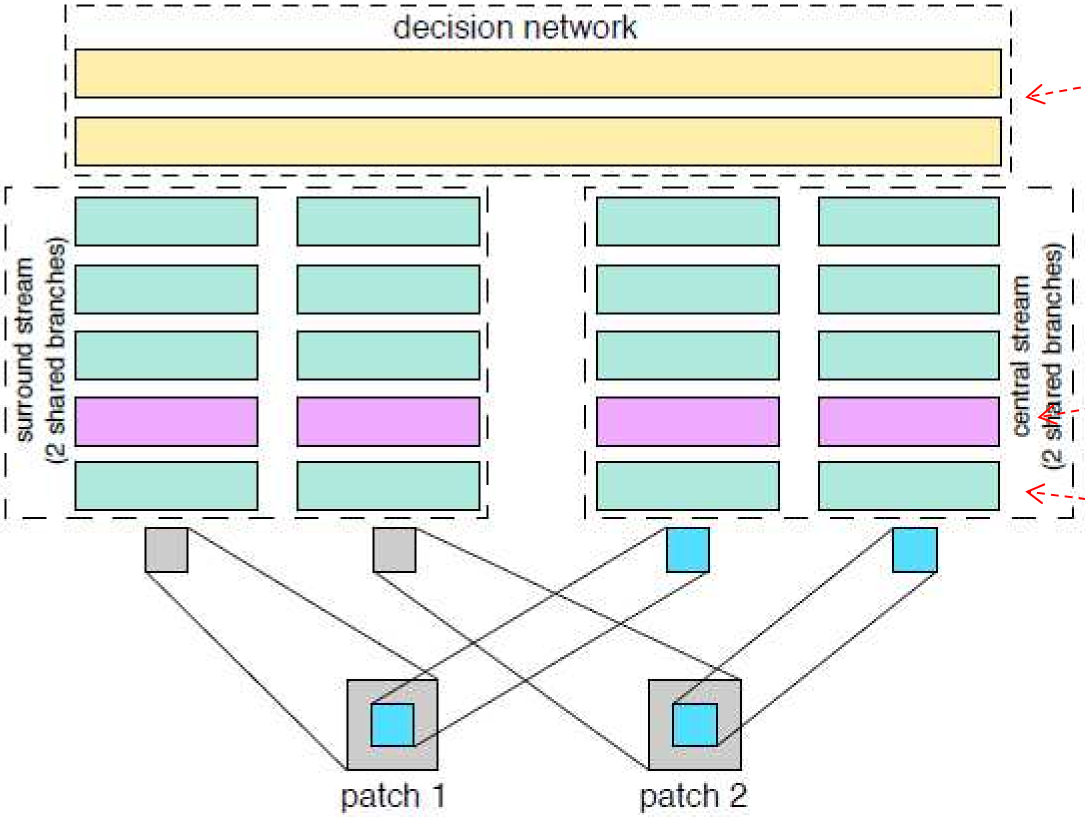
Three basic network architectures

- 1) 2-Channel, 2) Siamese, 3) Pseudo-Siamese



Extended Siamese network

- **<u>A central-surround two-stream network</u>** that uses a siamese-type architecture to process each stream



Fully connected layer

Max pooling

Conv+ReLU

Learning of Similarity Network

- **Optimization**
  - Optimizing with stochastic gradient descent (SGD) for the objective with hinge-based loss term + L2-norm regularization

$$\min_{w} \frac{\lambda}{2}\|w\|_2 + \sum_{i=1}^{N} \max(0, 1 - y_i o_i^{net})$$

1 (for positive samples) or
-1 (for negative samples)

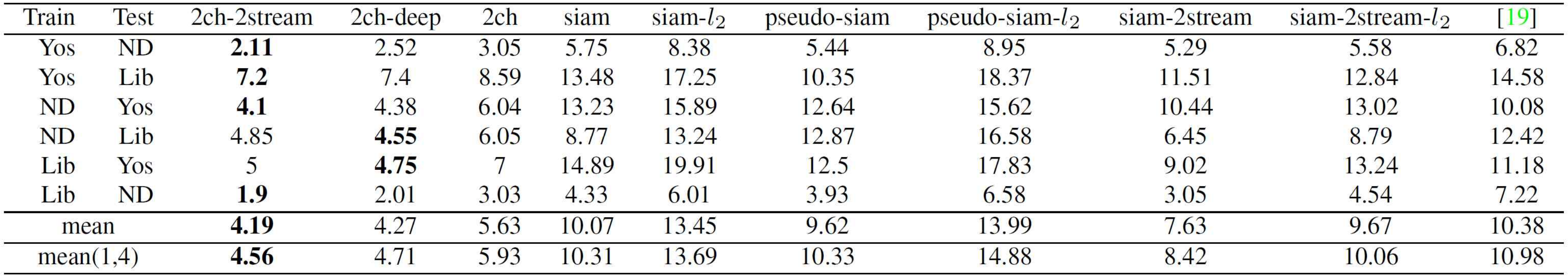
Network outputs

- **Data Augmentation and preprocessing**
  - To avoid overfitting, they augment training data by 1) flipping patches pairs horizontally and vertically and 2) rotating to 90, 180, 270 degrees.

Experimental Details

- **Training data**
  - *The patches are scale and orientation normalized.*
  - Three dataset: Yosemite, Notre Dame, and Liberty
    - 500,000 ground-truth feature pairs for each dataset, with equal number of positive (correct) and negative (incorrect) matches.
    - Each of the subsets was generated using actual correspondences obtained via multi-view stereo depth maps.

Experimental Results



Train	Test	2ch-2stream	2ch-deep	2ch	siam	siam-l_2	pseudo-siam	pseudo-siam-l_2	siam-2stream	siam-2stream-l_2	[19]
Yos	ND	**2.11**	2.52	3.05	5.75	8.38	5.44	8.95	5.29	5.58	6.82
Yos	Lib	**7.2**	7.4	8.59	13.48	17.25	10.35	18.37	11.51	12.84	14.58
ND	Yos	**4.1**	4.38	6.04	13.23	15.89	12.64	15.62	10.44	13.02	10.08
ND	Lib	4.85	**4.55**	6.05	8.77	13.24	12.87	16.58	6.45	8.79	12.42
Lib	Yos	5	**4.75**	7	14.89	19.91	12.5	17.83	9.02	13.24	11.18
Lib	ND	**1.9**	2.01	3.03	4.33	6.01	3.93	6.58	3.05	4.54	7.22
mean		**4.19**	4.27	5.63	10.07	13.45	9.62	13.99	7.63	9.67	10.38
mean(1,4)		**4.56**	4.71	5.93	10.31	13.69	10.33	14.88	8.42	10.06	10.98

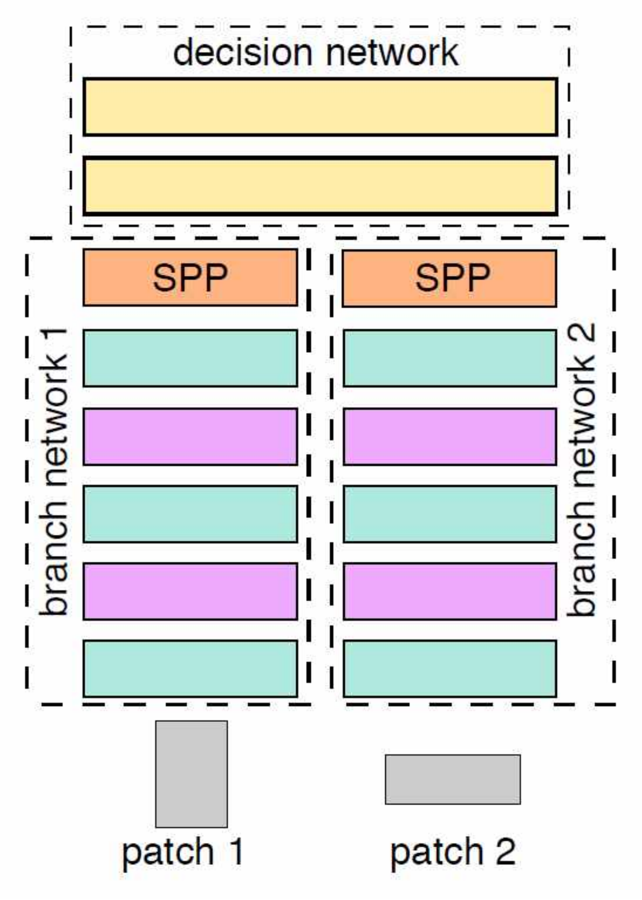
- 2-channel & 2 stream network is the best.
- Decision network works better than simple L2 distance.

Key idea: learning the pooling regions for defining feature descriptors based on sparsity (Hand-crafted descriptors)

[19] Learning Local Feature Descriptors Using Convex Optimisation, IEEE TPAMI 2014

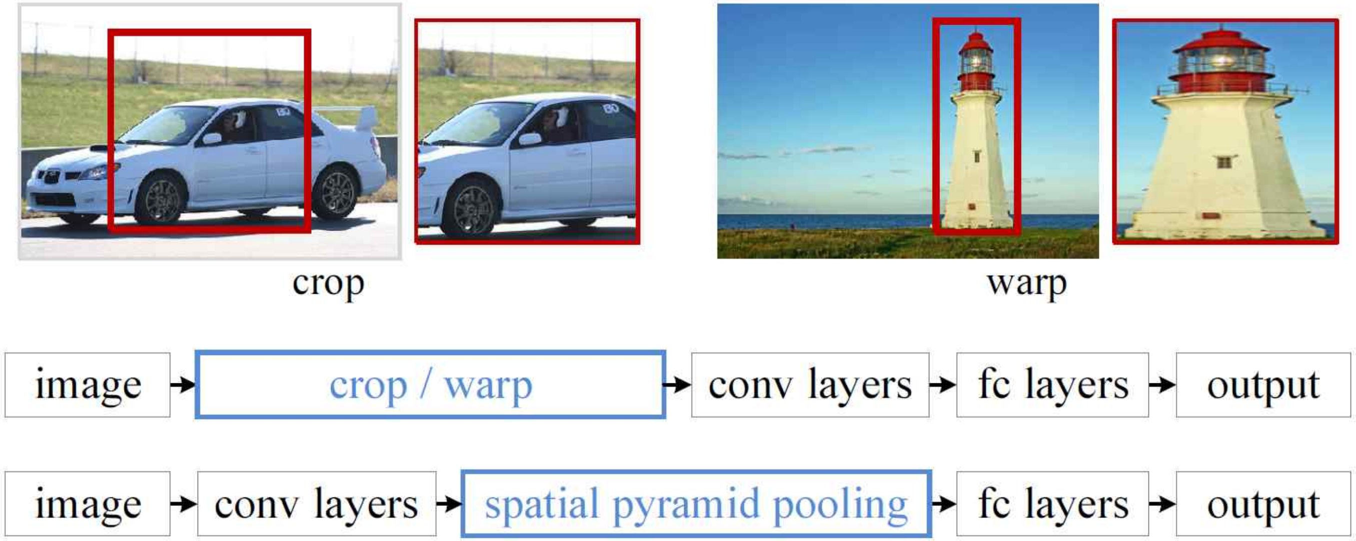
Similarity Network using SPP

- **Putting spatial pyramid pooling (SPP) on the top of branch networks**
  - Top decision layer has an input of fixed dimensionality for any size of the input patches.



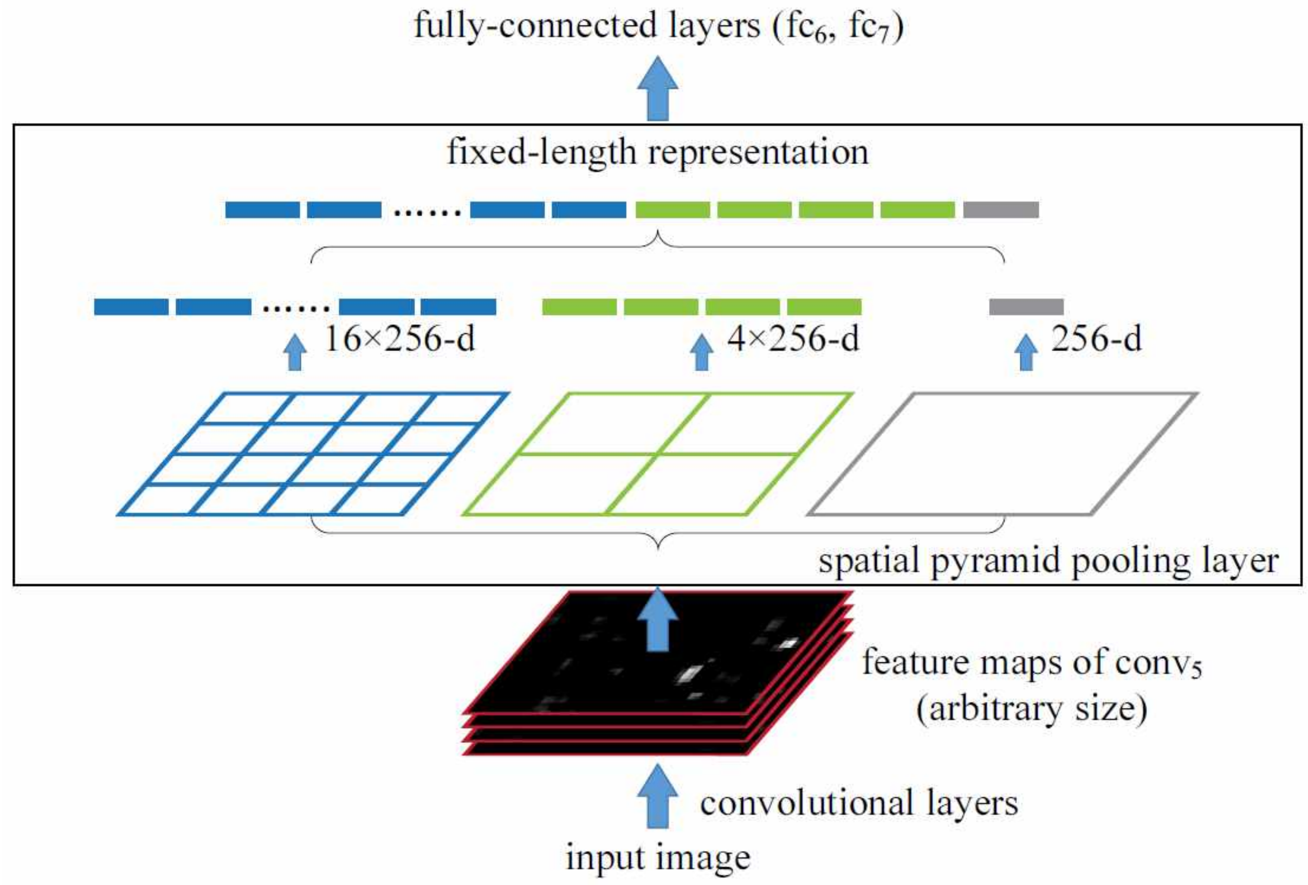
Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, TPAMI 2015

- Addresses the implementation issue that CNN takes an input with a fixed size only.


crop
warp
image → crop / warp → conv layers → fc layers → output
image → conv layers → spatial pyramid pooling → fc layers → output

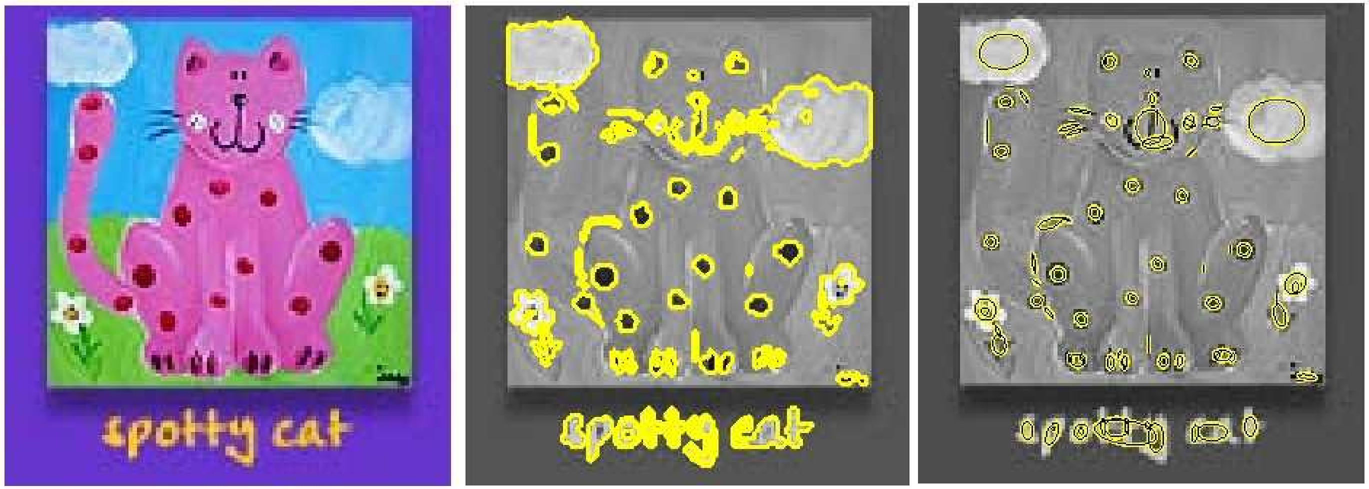
Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, TPAMI 2015

- Multiple responses are concatenated from spatial pyramid pooling layers.



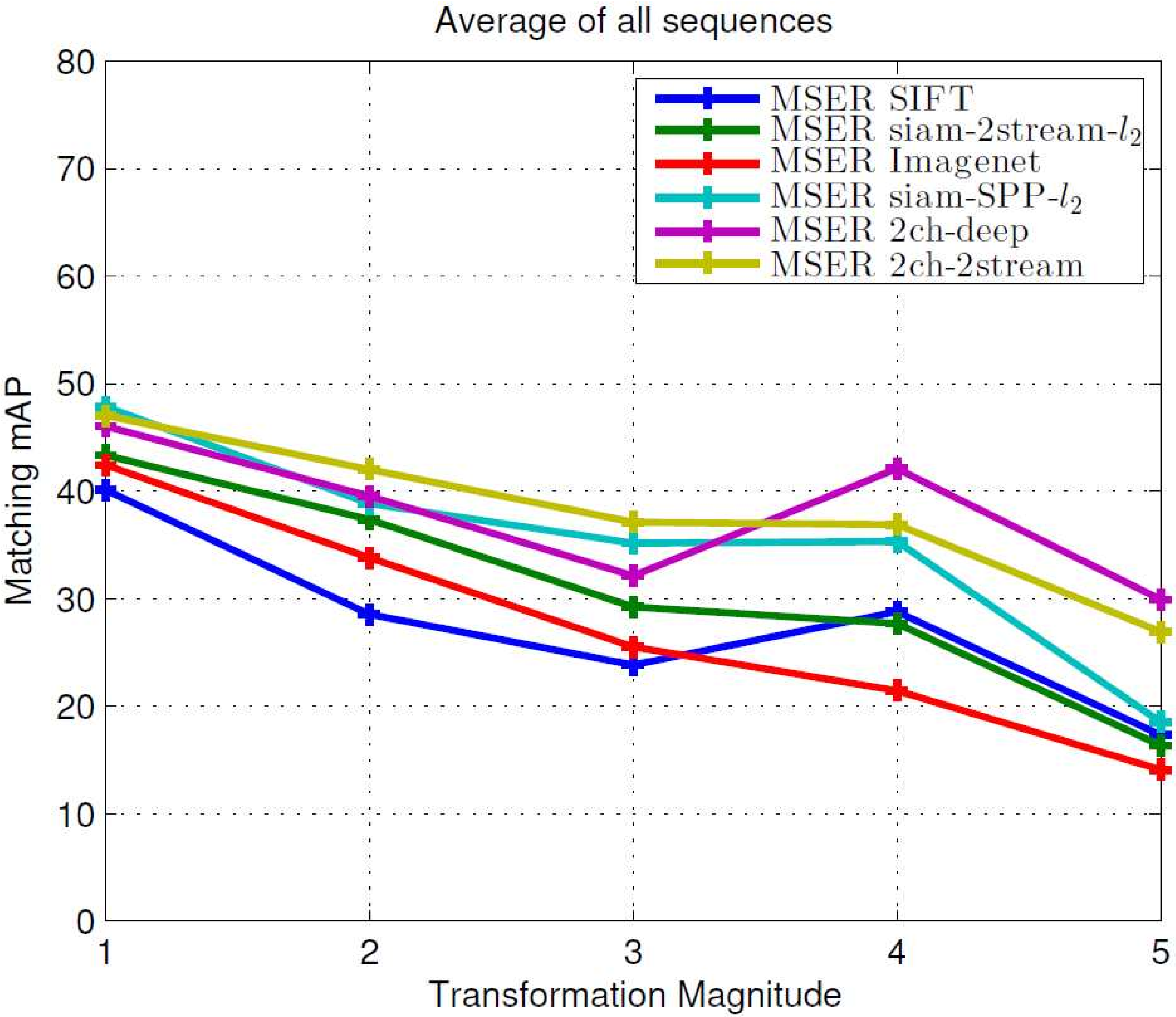
Similarity Network using SPP

1. Use the ellipses detected by MSER (Maximally stable extremal regions) for interest points.
2. These ellipses are used as inputs for the similarity network using SPP.



MSER results from http://www.vlfeat.org/overview/mser.html

Local descriptors performance evaluation



Concluding Remarks

- 2-channel 2-stream network produce the best results

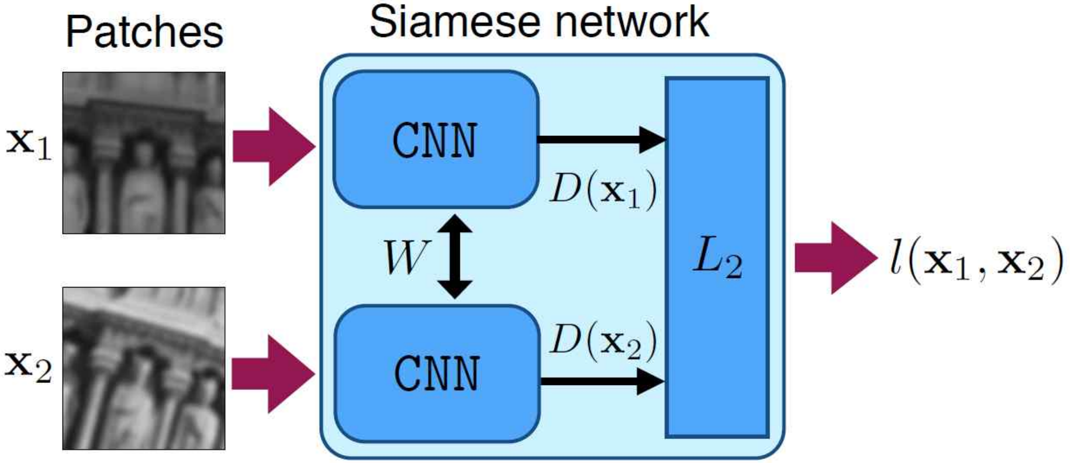
→ Future work: Accelerating the evaluation of this network

- 2-stream multi-resolution models and SPP based models consistently improve the descriptor quality.

- Learning with a larger training set may improve the performance of the proposed method.

Discriminative Learning of Deep Convolutional Feature Point Descriptors, ICCV 2015

Schematic of a Siamese network, where pairs of input patches are processed by two copies of the same CNN.



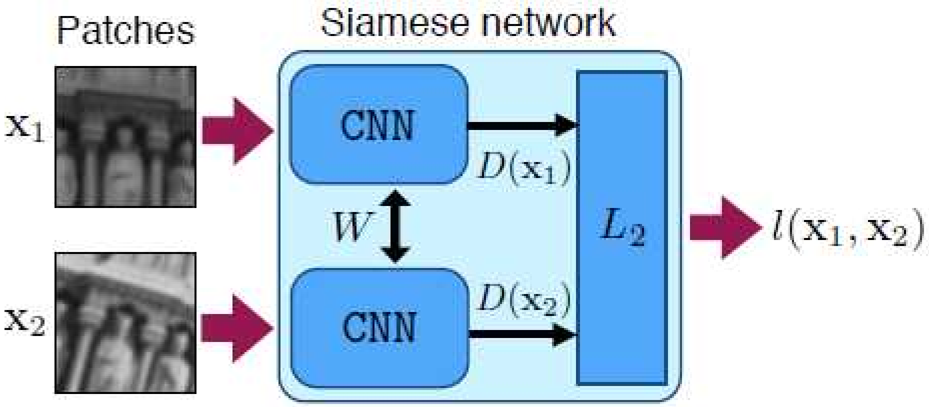
Note) This is almost similar to CVPR 2015 paper, except using L2 distance

Positive examples

$$l(\mathbf{x}_1, \mathbf{x}_2) = \begin{cases} \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2, & p_1 = p_2 \\ \max(0, C - \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2), & p_1 \neq p_2 \end{cases}$$

Negative examples

Discriminative Learning of Deep Convolutional Feature Point Descriptors, ICCV 2015



Complexity matters!

Patch-wise similarity measure is extremely slow.

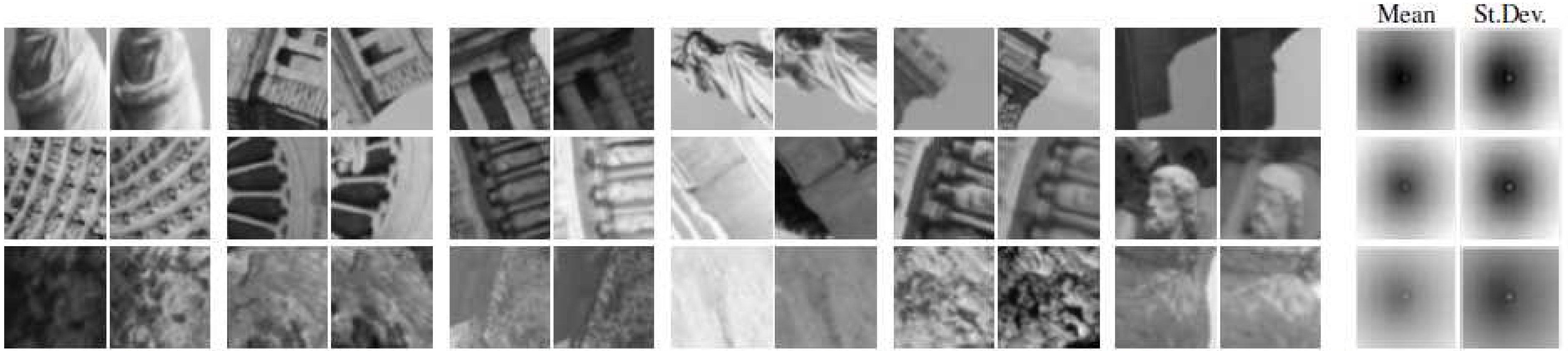


Figure 3: Pairs of corresponding samples from the MVS dataset. Top: 'Liberty' (LY). Middle: 'Notre Dame' (ND). Bottom: 'Yosemite' (YO). Right: we compute the pixel difference between corresponding patches on each set and show their mean/std.

PART 1.2: LEARNING-BASED DESCRIPTORS

UNIVERSAL CORRESPONDENCE NETWORK

C. B. Choy, Y. Gwak, and S. Savarese, “Universal Correspondence Network,” NIPS, 2016
M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial Transformer Networks,” NIPS 2015

Spatial Transformer Network, NIPS 2015

- **Goal**: dealing with spatial transformation in an end-to-end training framework

- Interleaving convolutional layers with max-pooling layers allows translation invariance.

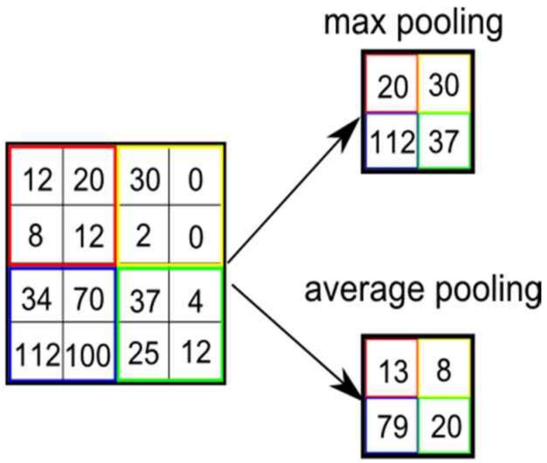
+ Exceptionally effective

- Pooling is simplistic.

- Only small invariances per pooling layer

- Limited spatial transformation

- Pools across entire image



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

max pooling

20	30
112	37

average pooling

13	8
79	20

- Can we do better?

M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial Transformer Networks,” NIPS 2015

Slide Courtesy from M. Jaederberg

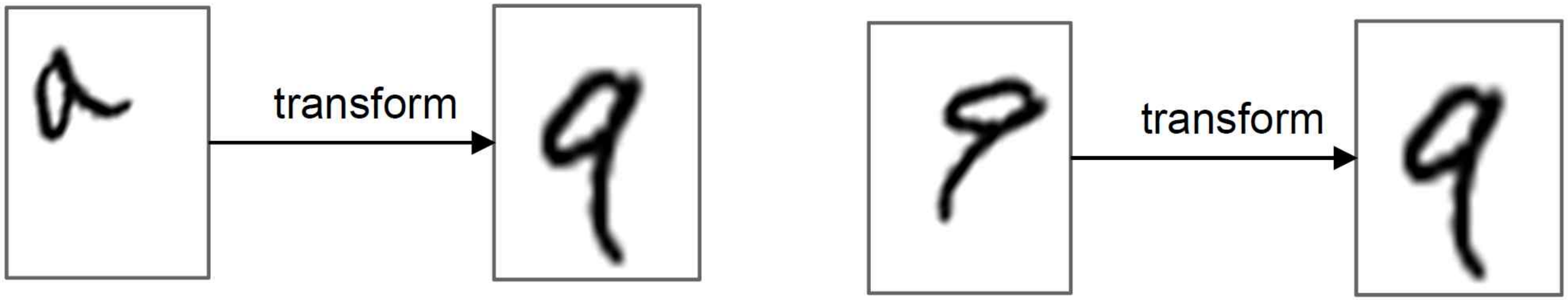
Conditional Spatial Warping

- Conditional on input feature map, spatially warp image.

+ Transforms data to a space expected by subsequent layers

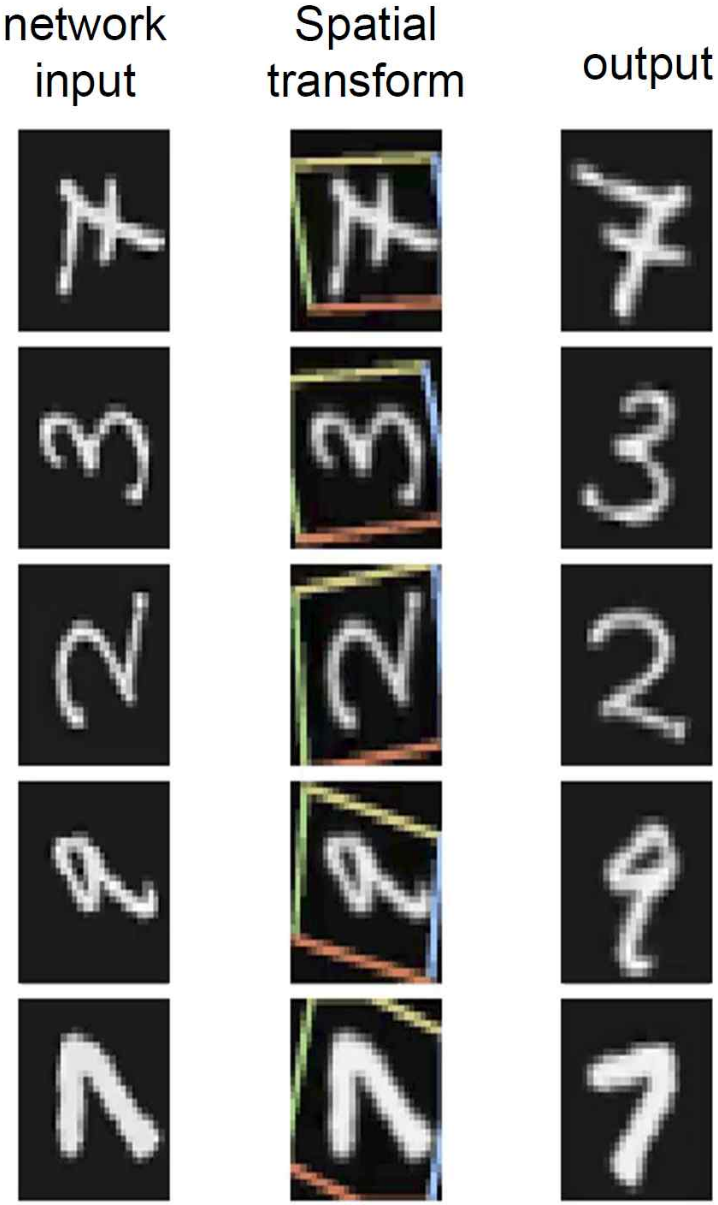
+ Intelligently select features of interest (attention)

+ Invariant to more generic warping



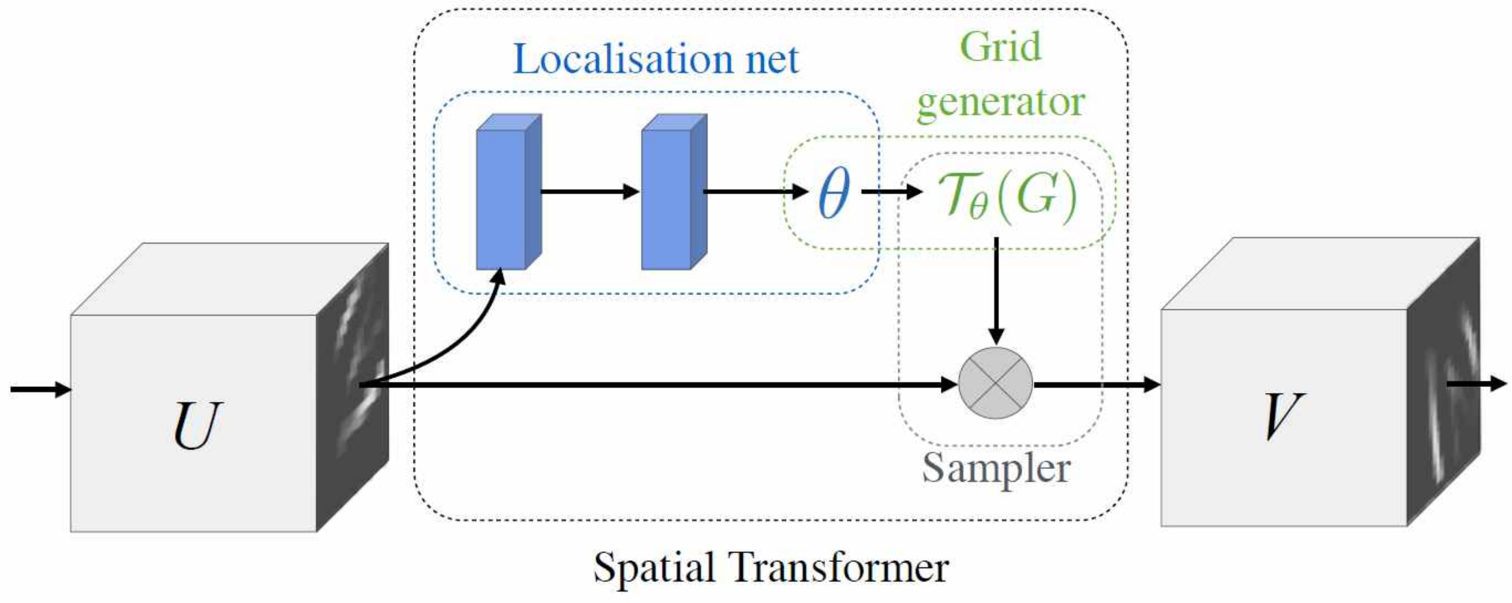
Slide Courtesy from M. Jaederberg

Conditional Spatial Warping



Slide Courtesy from M. Jaederberg

A differentiable module for spatially transforming data, conditional on the data itself

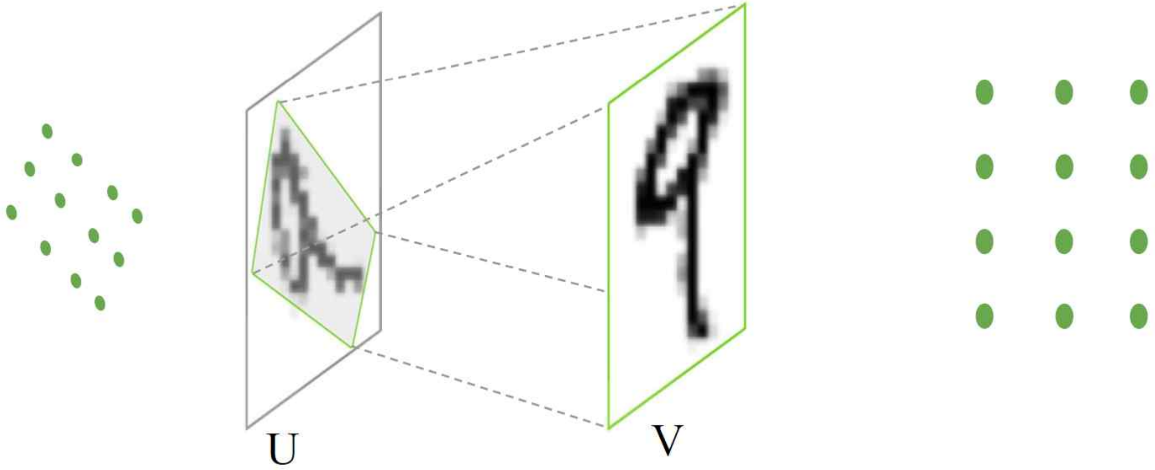


Slide Courtesy from M. Jaederberg

Sampling Grid

- Warp regular grid by an affine transformation

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

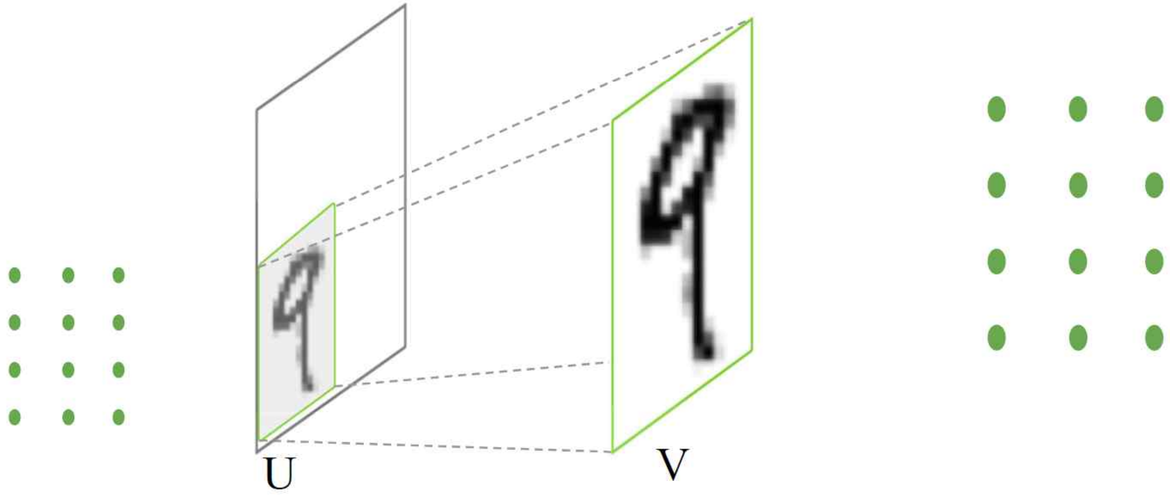


Slide Courtesy from M. Jaederberg

Sampling Grid

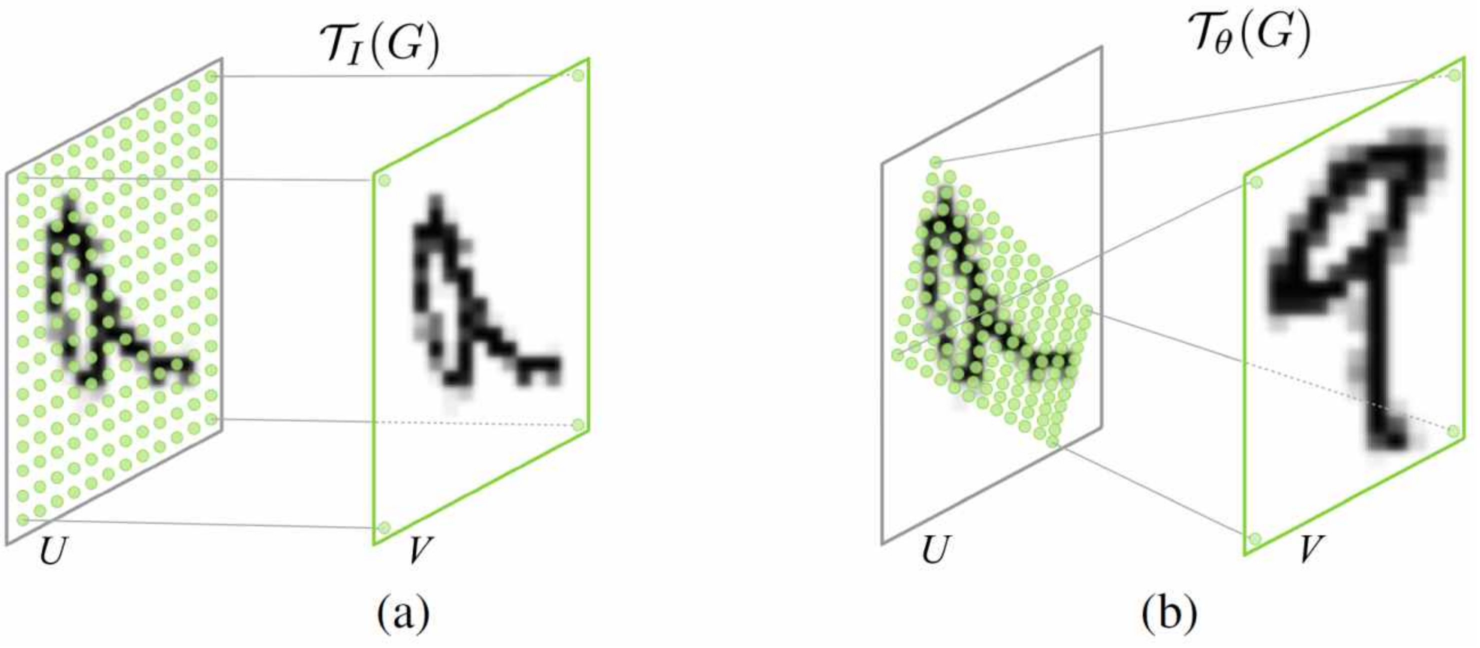
- Warp regular grid by an affine transformation (Attention model)

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



Slide Courtesy from M. Jaederberg

Conditional Spatial Warping



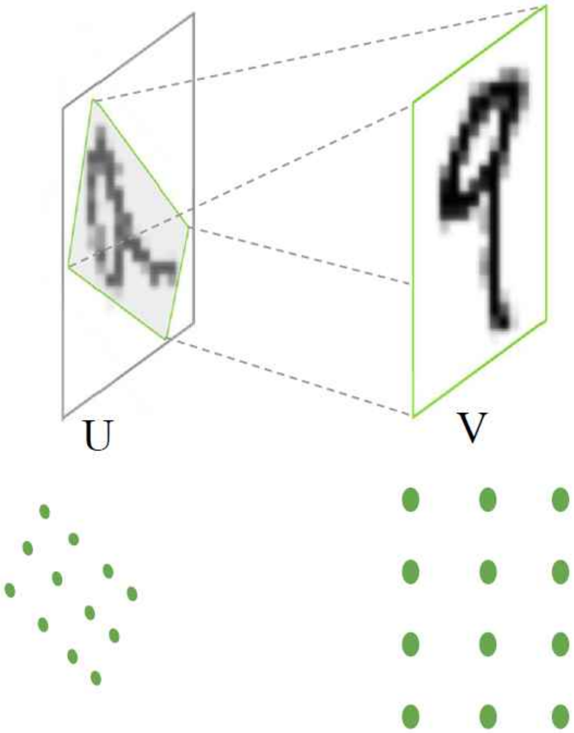
Identity transformation

Affine transformation

Slide Courtesy from M. Jaederberg

Sampler

- Sample input feature map U to produce output feature map V (i.e. texture mapping)



e.g. for bilinear interpolation:

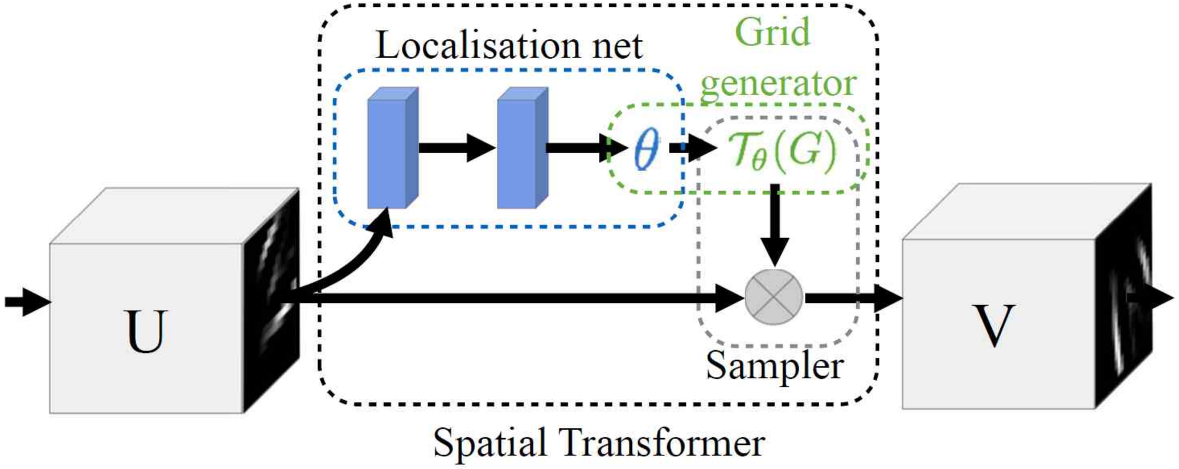
$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|)$$

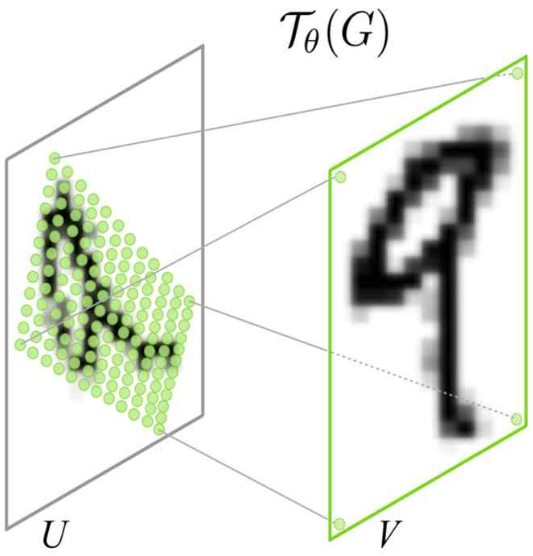
and gradients are defined to allow backpropagation, eg:

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

Slide Courtesy from M. Jaederberg

A differentiable module for spatially transforming data, conditional on the data itself

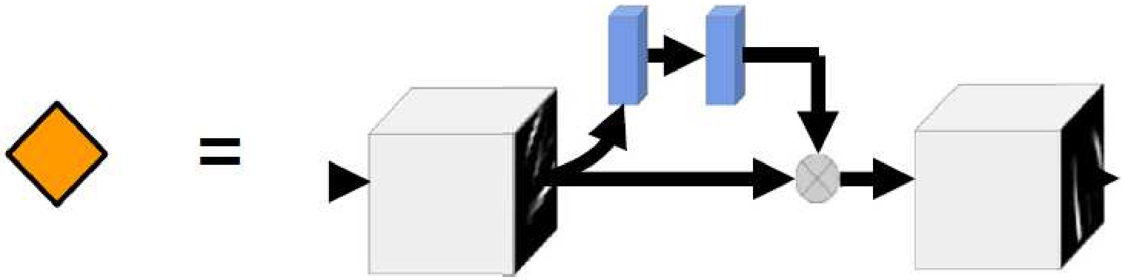




Slide Courtesy from M. Jaederberg

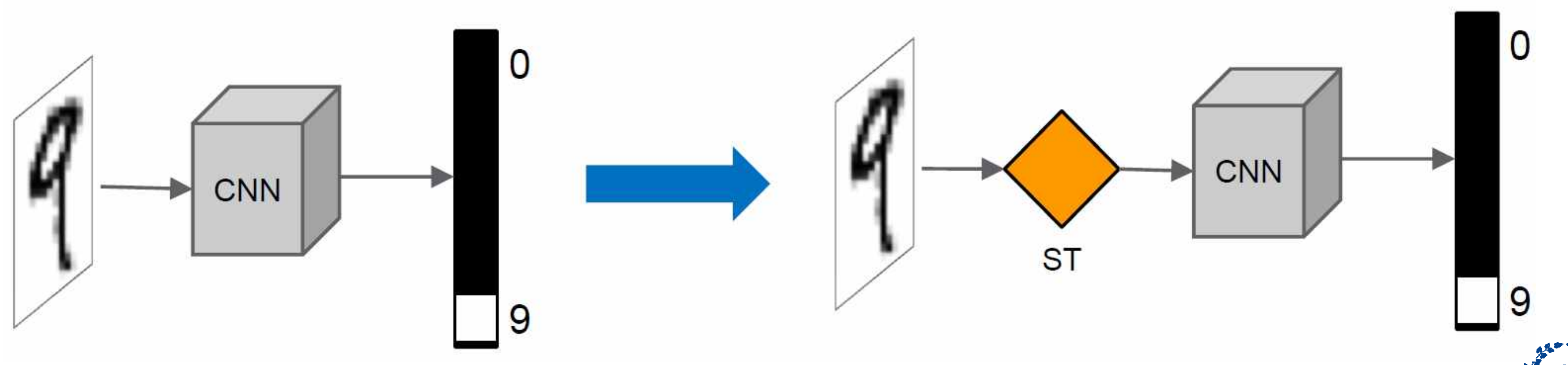
Spatial Transformer Networks

- Spatial Transformers is fully *differentiable*, and so can be inserted at any point in a feed forward network and trained by back propagation



Example:

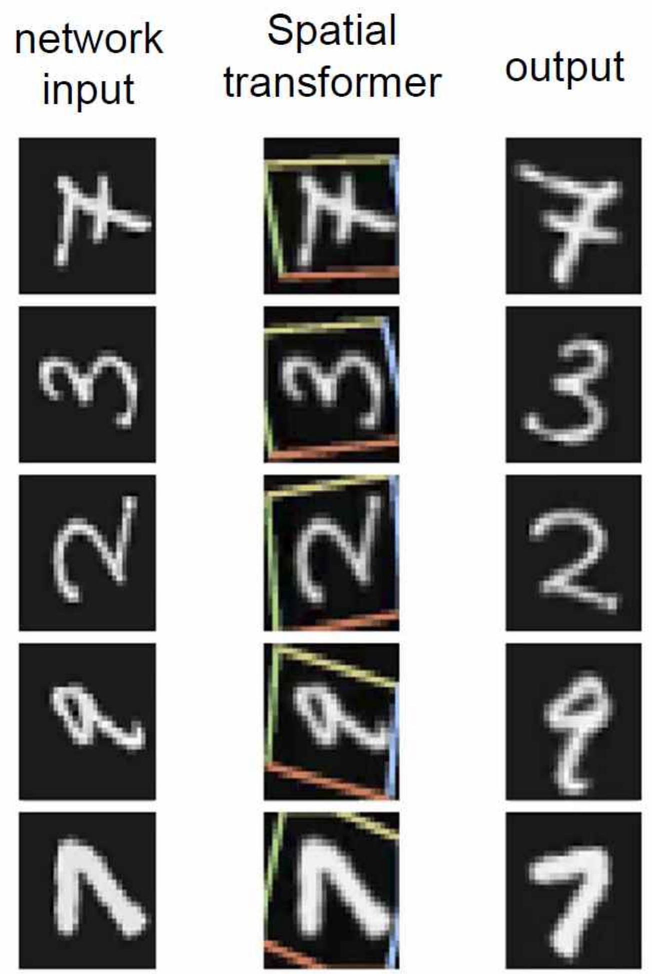
- digit classification, loss: cross-entropy for 10 way classification



Slide Courtesy from M. Jaederberg

Task: classify MNIST digits

- Training and test randomly rotated by (+/- 90°)
- Fully connected network with affine ST on input



Performance:

- FCN 2.1
- CNN 1.2
- ST-FCN 1.2
- ST-CNN 0.7

Slide Courtesy from M. Jaederberg

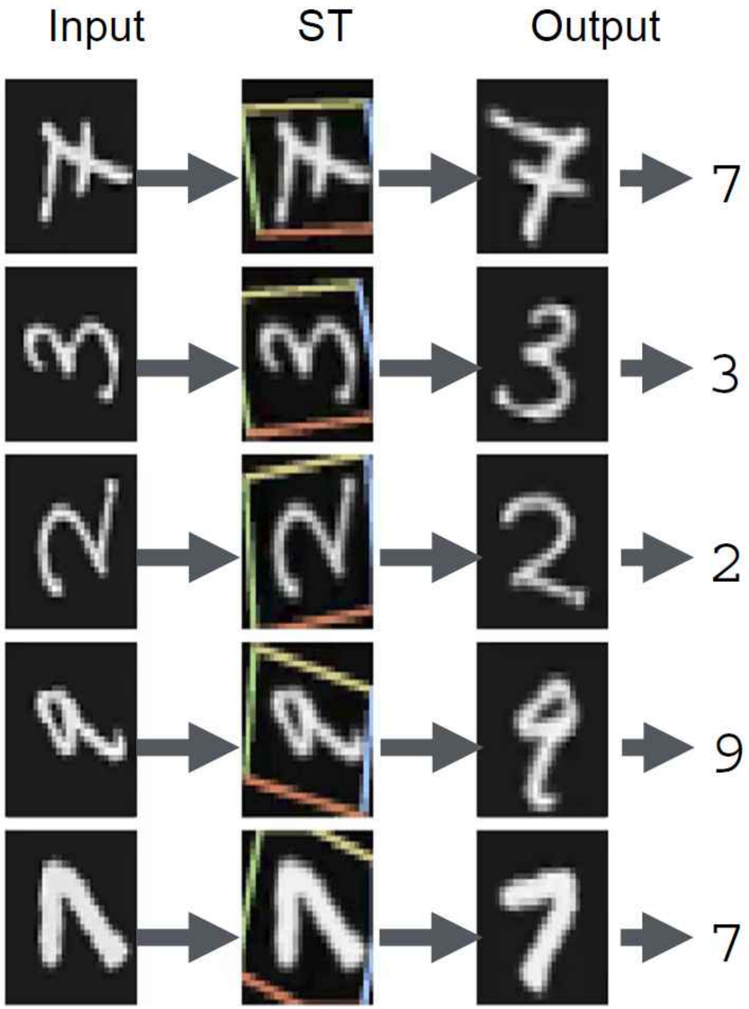
Generalizations 1: transformations

- Affine transformation – 6 parameters
- Projective transformation – 8 parameters
- Thin plate spline transformation
- Etc
- Any transformation where parameters can be regressed

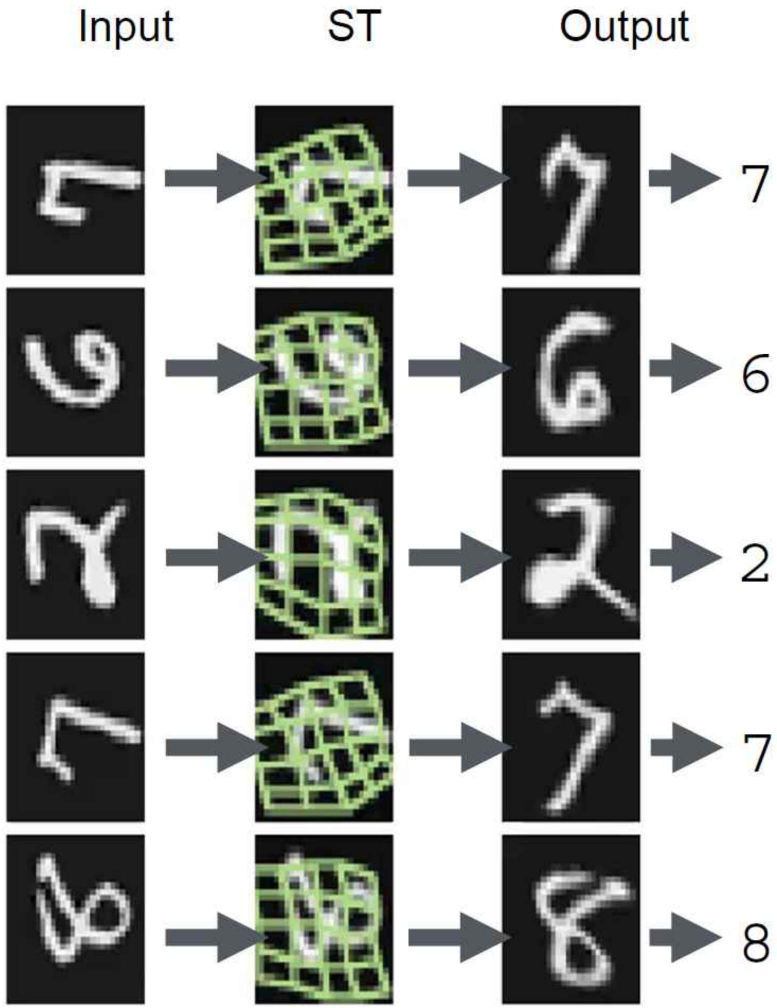
Slide Courtesy from M. Jaederberg

Rotated MNIST

ST-FCN Affine

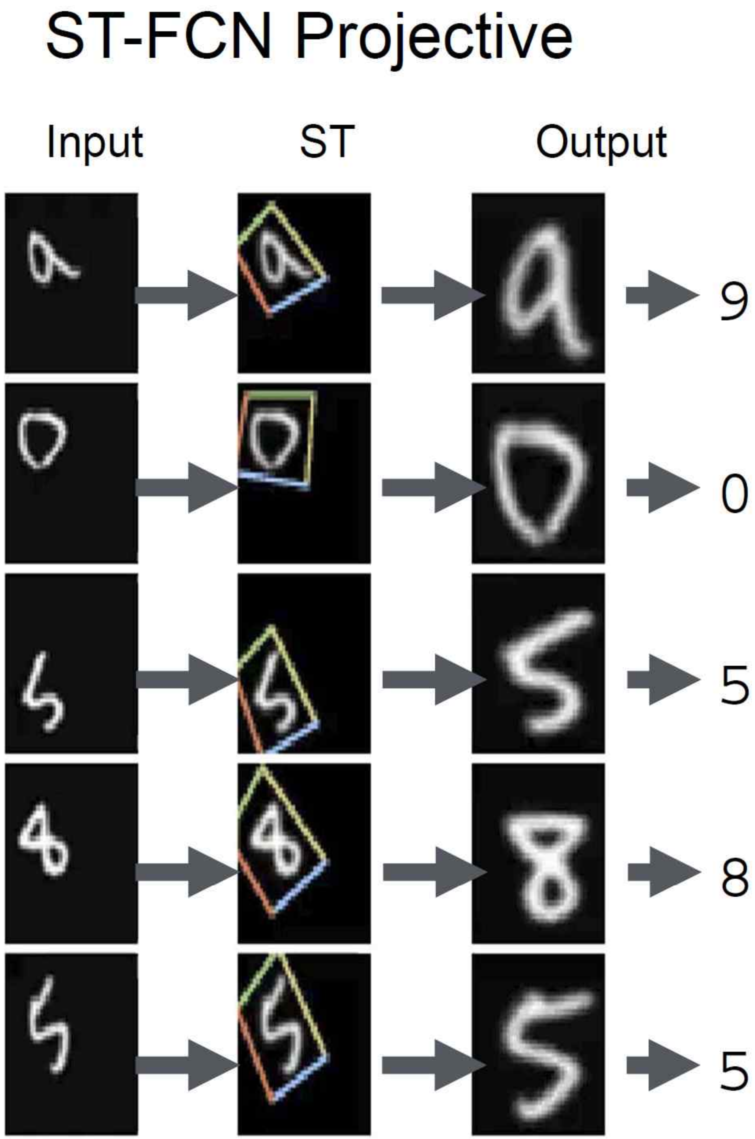


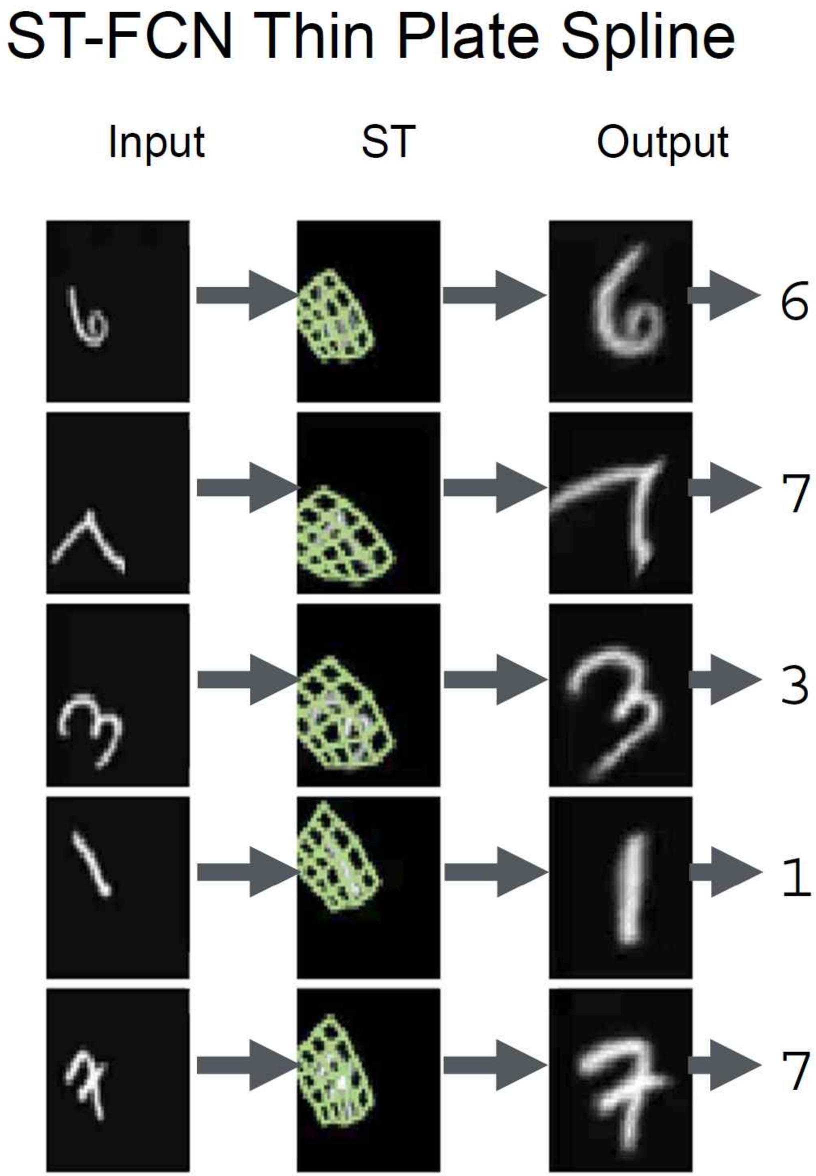
ST-FCN Thin Plate Spline



Slide Courtesy from M. Jaederberg

Rotated, Translated & Scaled MNIST

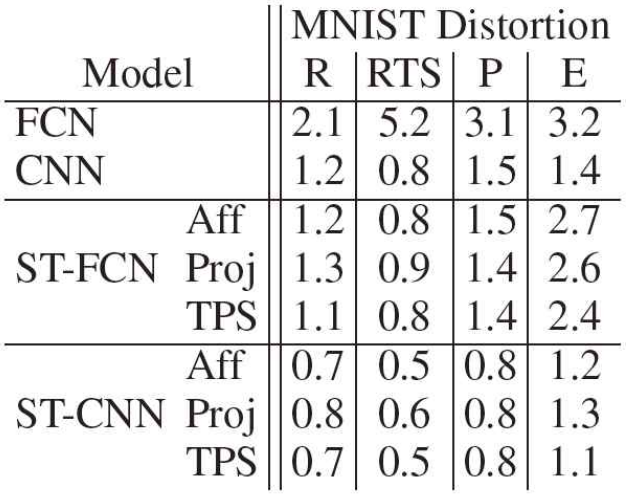




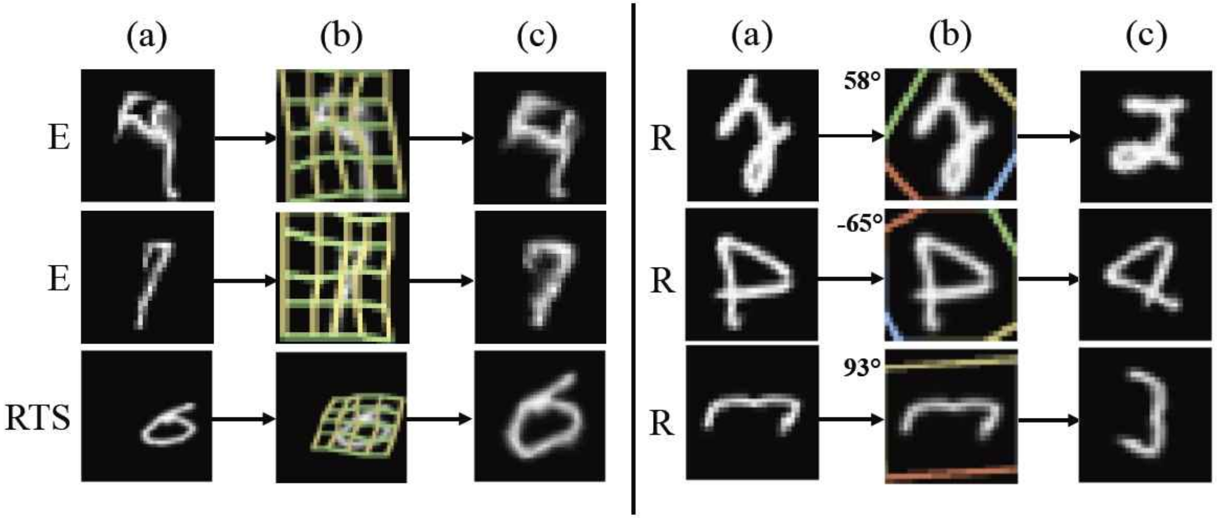
Slide Courtesy from M. Jaederberg

Objective Performance

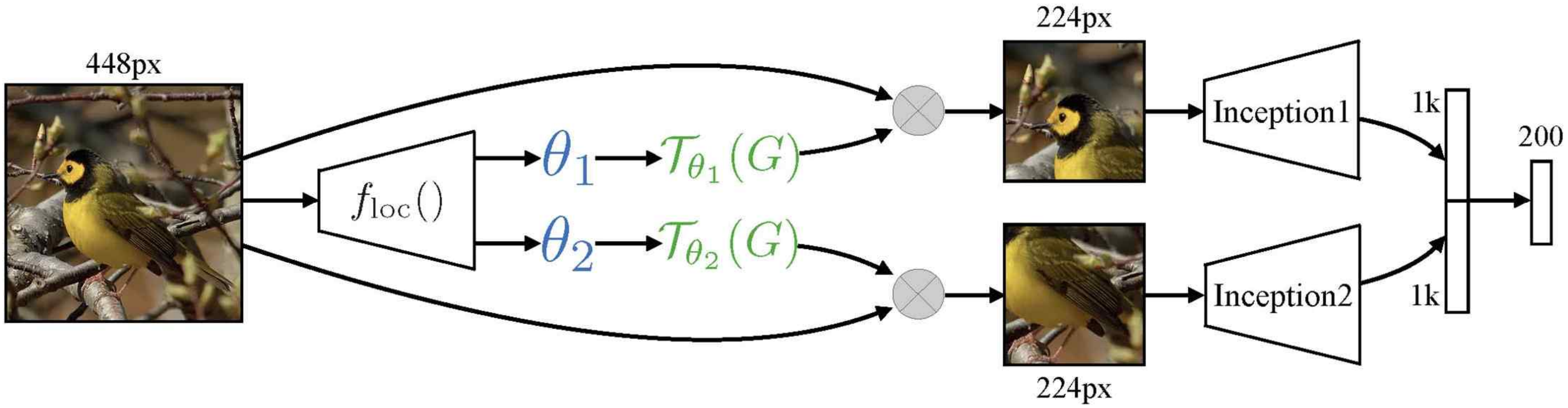
- The percentage errors for different models on different distorted MNIST datasets



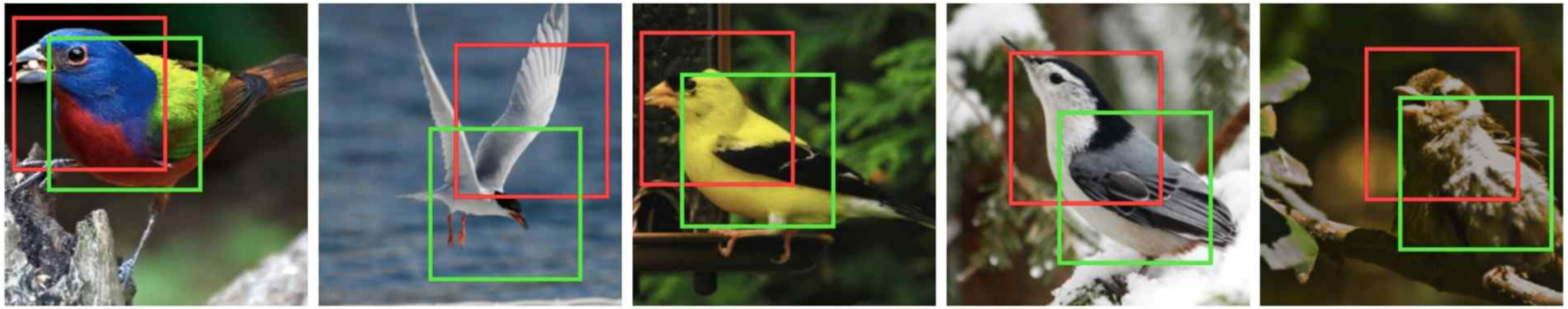
Model		MNIST Distortion			
		R	RTS	P	E
FCN		2.1	5.2	3.1	3.2
CNN		1.2	0.8	1.5	1.4
ST-FCN	Aff	1.2	0.8	1.5	2.7
	Proj	1.3	0.9	1.4	2.6
	TPS	1.1	0.8	1.4	2.4
ST-CNN	Aff	0.7	0.5	0.8	1.2
	Proj	0.8	0.6	0.8	1.3
	TPS	0.7	0.5	0.8	1.1



App: Fine Grained Visual Categorization



- Pre-train inception networks on ImageNet
- Train spatial transformer network on fine grained multi-way classification



Slide Courtesy from M. Jaederberg

Summary of STN

- Spatial Transformers allow dynamic, conditional cropping and warping of images/feature maps.

- Can be constrained and used as very fast attention mechanism.

- Spatial Transformer Networks localize and rectify objects automatically. Achieve state of the art results.

- Can be used as a generic localization mechanism which can be learnt with back-propagation.

Slide Courtesy from M. Jaederberg

Universal Correspondence Network (UCN), NIPS 2016

- **Hand-crafted descriptors**
  - Count on local image properties such as image gradient.
  - Different descriptors are used for various correspondence applications
    - SIFT, SURF: sparse structure from motion
    - DAISY, Deformable Spatial Pyramid (DSP): dense matching
    - SIFT Flow, FlowWeb: semantic matching

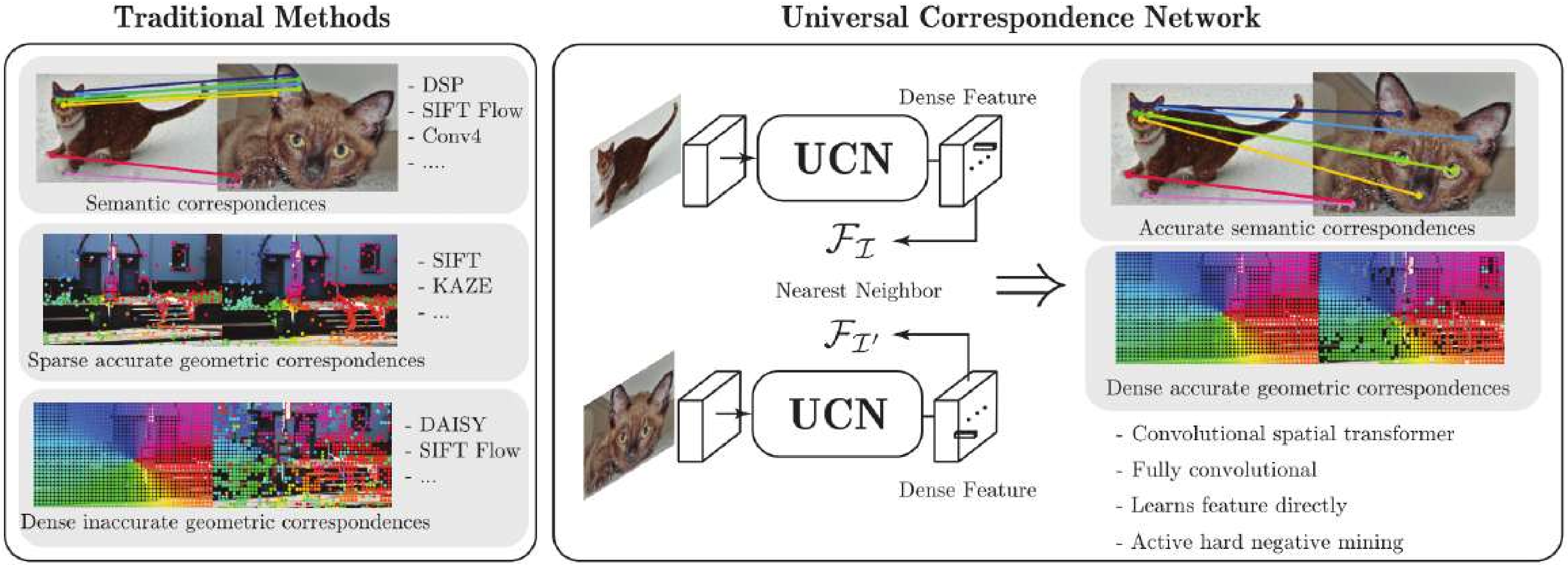
- **Existing learning-based descriptors**
  - Typically deal with patch-wise similarity using Siamese network.
  - It is well-known that CNN is invariant to scale and translation thanks to convolution and pooling layers.
  - However, handling variations with data augmentation or explicit network structure yields higher accuracy!

    → Spatial transformer network (STN, NIPS 2015)

Universal Correspondence Network (UCN)

- The UCN learns a **metric space** for geometric correspondences, dense trajectories or semantic correspondences.

- Existing learning-based descriptors using patch-similarity require $O(n^2)$ feed-forward passes where n: # of patches, while UCN use only $O(n)$.
  - Note that this is very similar to the fast version of MC-CNN.


Traditional Methods
- DSP
- SIFT Flow
- Conv4
- ....
Semantic correspondences
- SIFT
- KAZE
- ...
Sparse accurate geometric correspondences
- DAISY
- SIFT Flow
- ...
Dense inaccurate geometric correspondences
Universal Correspondence Network
Dense Feature
UCN
$\mathcal{F}_{\mathcal{I}}$
Nearest Neighbor
$\mathcal{F}_{\mathcal{I}'}$
UCN
Dense Feature
Accurate semantic correspondences
Dense accurate geometric correspondences
- Convolutional spatial transformer
- Fully convolutional
- Learns feature directly
- Active hard negative mining

Three Key Contributions

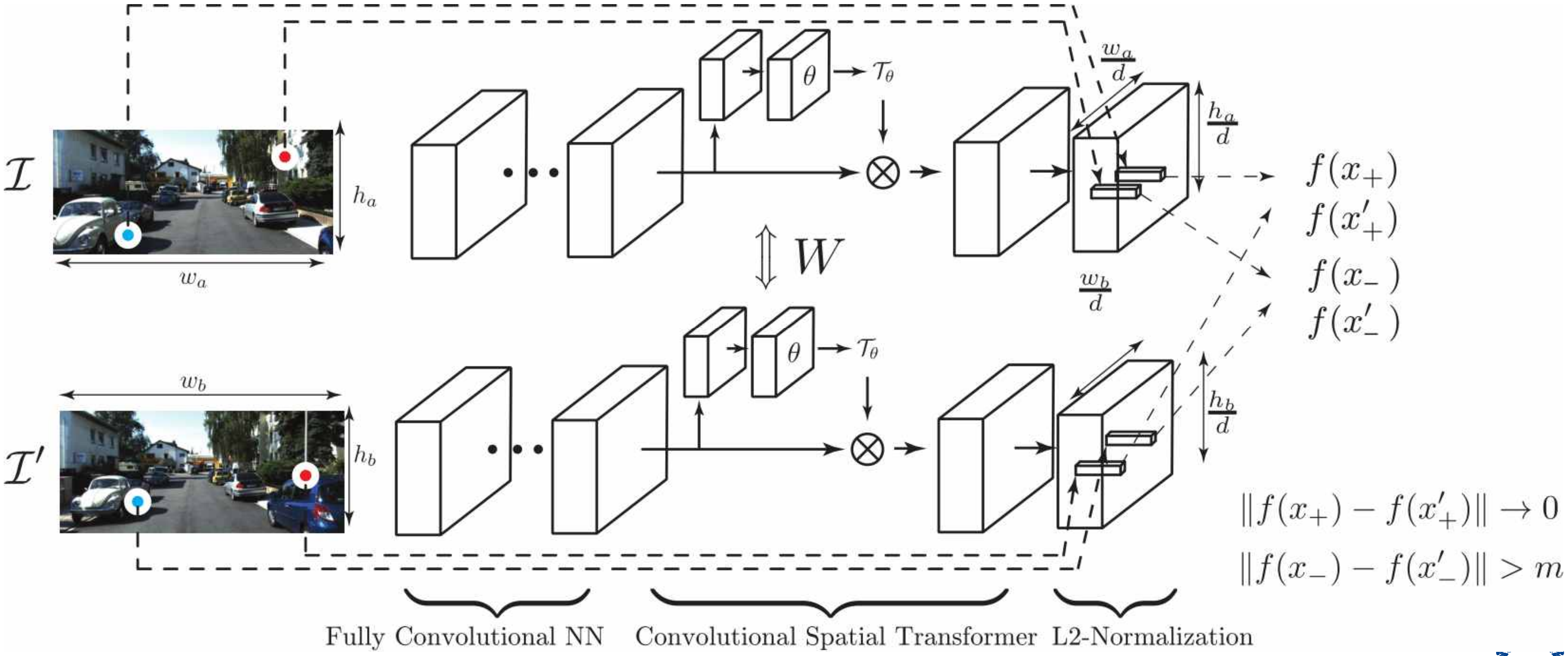
1. Deep metric learning with a constrastive loss for learning a feature representation that is optimized for the given correspondence task.

2. Fully convolutional network with fast active hard negative mining.

3. Fully convolutional spatial transformer for patch normalization, by incorporating spatial transformer network (STN) in their network.

Network Architecture

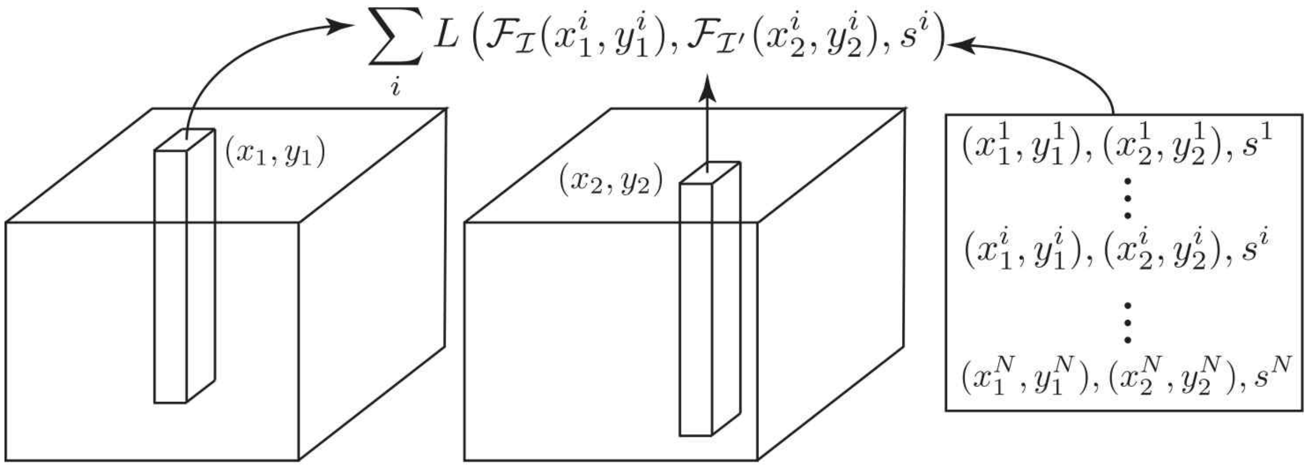
- **Fully convolutional NN**: convolutions, pooling, and nonlinearities (ReLU)
- **Convolutional spatial transformer**: deal with geometric variations
- **Channel-wise L2 normalization**: is similar to SIFT
- **Correspondence contrastive loss**: is used for an effective learning



Correspondence Contrastive Loss

- Generalized form of contrastive loss
  - Key idea: use a set of all patches, NOT just a single patch.

$$L = \frac{1}{2N}\sum_{i}^{N} s_i\|\mathcal{F}_{\mathcal{I}}(\mathbf{x}_i) - \mathcal{F}_{\mathcal{I}'}({\mathbf{x}_i}')\|^2 + (1-s_i)\max(0, m - \|\mathcal{F}_{\mathcal{I}}(\mathbf{x}) - \mathcal{F}_{\mathcal{I}'}({\mathbf{x}_i}')\|)^2$$


$\sum_i L\left(\mathcal{F}_{\mathcal{I}}(x_1^i, y_1^i), \mathcal{F}_{\mathcal{I}'}(x_2^i, y_2^i), s^i\right)$
(x_1, y_1)
(x_2, y_2)
$(x_1^1, y_1^1), (x_2^1, y_2^1), s^1$
$\vdots$
$(x_1^i, y_1^i), (x_2^i, y_2^i), s^i$
$\vdots$
$(x_1^N, y_1^N), (x_2^N, y_2^N), s^N$

Note) Compare with the following contrastive loss used in ICCV 2015 paper

$$l(\mathbf{x}_1, \mathbf{x}_2) = \begin{cases} \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2, & p_1 = p_2 \\ \max(0, C - \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2), & p_1 \neq p_2 \end{cases}$$

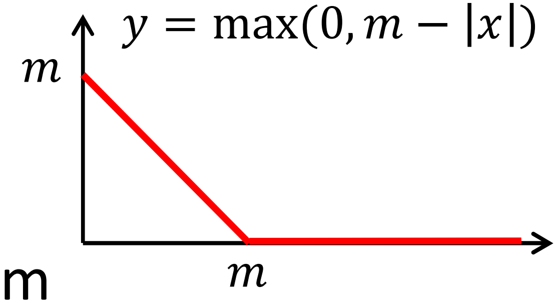
Hard Negative Mining

$$L = \frac{1}{2N}\sum_{i}^{N} s_i\|\mathcal{F}_{\mathcal{I}}(\mathbf{x}_i) - \mathcal{F}_{\mathcal{I}'}(\mathbf{x}_i{}')\|^2 + (1-s_i)\max(0, m - \|\mathcal{F}_{\mathcal{I}}(\mathbf{x}) - \mathcal{F}_{\mathcal{I}'}(\mathbf{x}_i{}')\|)^2$$

Positive samples

Negative samples

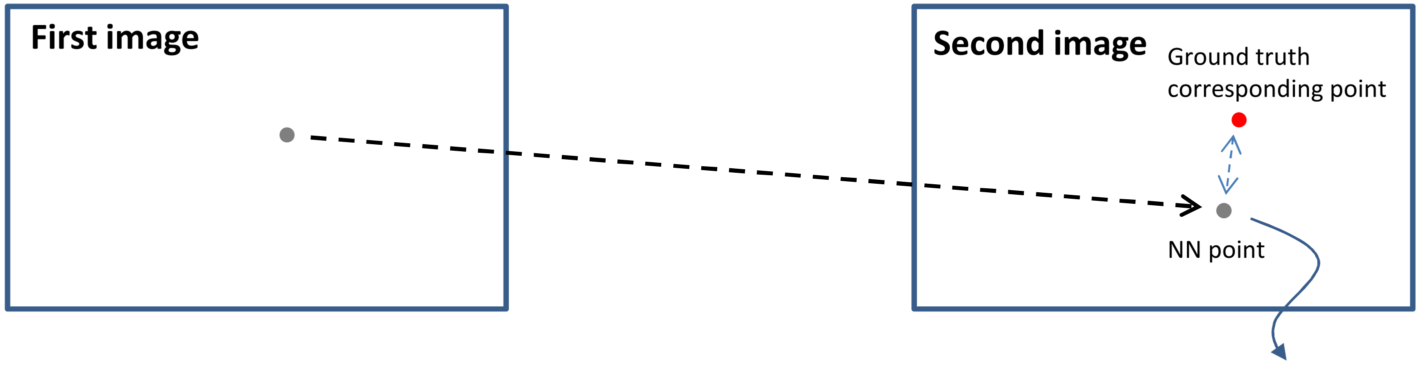
- The second term is active only when the distance between the feature are *smaller* than the margin m.



- So, random negative pairs do not contribute to training, since they are generally too far from each other.

Hard Negative Mining

- **Hard negative mining solution in UCN**
  1) Extract features in the first image
  2) Find the nearest neighbor (NN) in the second image
  3) Use as negative pairs NN candidates far from the ground truth



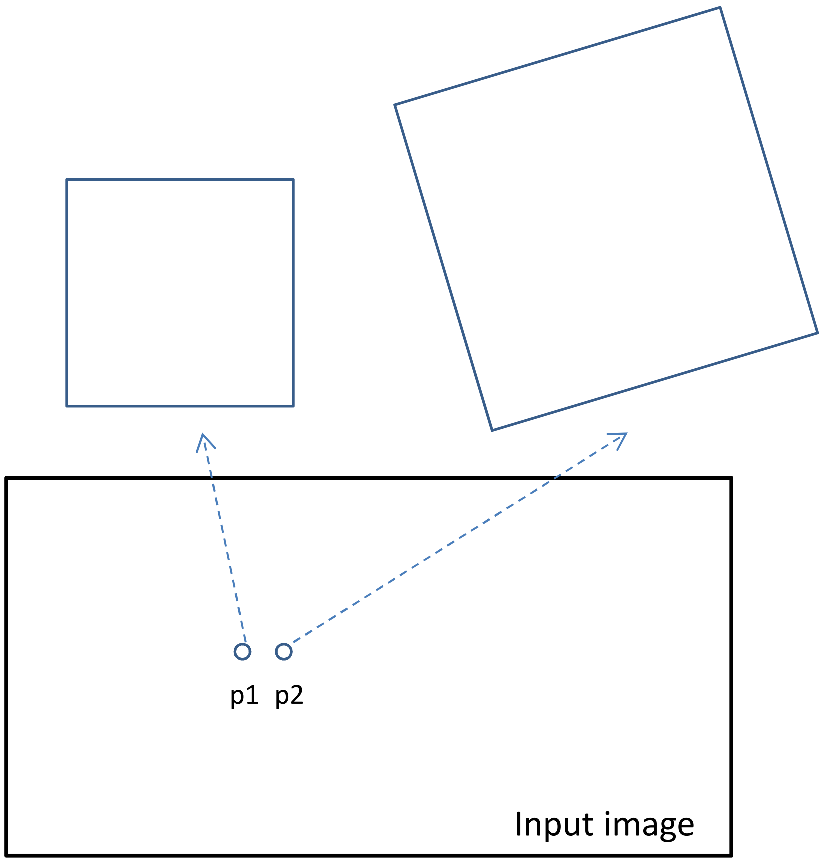
This is a negative sample!

Geometric Invariance in CNNs

- Suppose two adjacent pixels have **different** scales and rotations

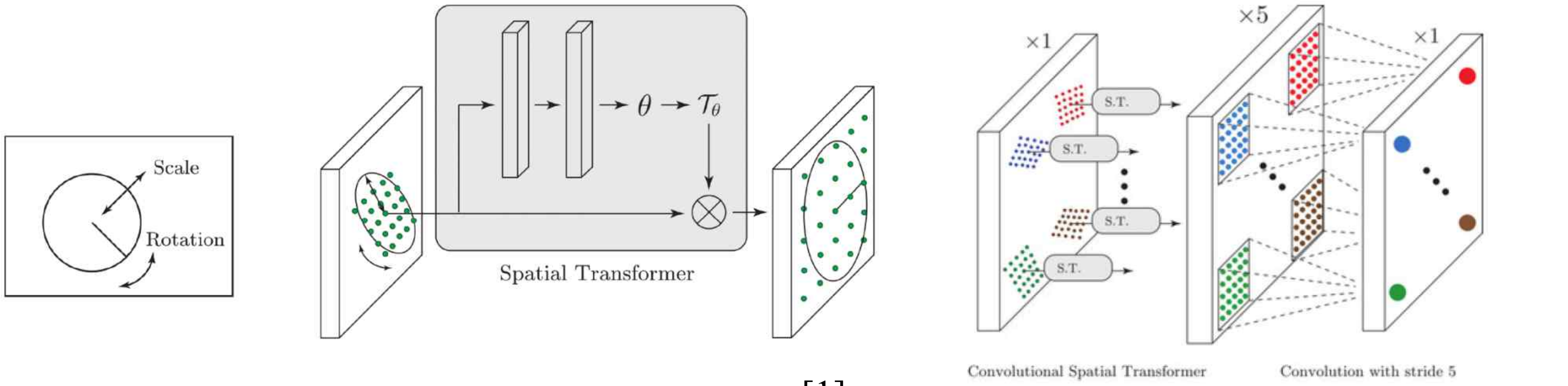
p1: scale = 1, rotation = 0
p2: scale = 1.5, rotation = 30

Problem: Patch size and orientation are different from all pixels
-> Convolutional kernel should be varying for each pixel, which is contradictory to conventional CNNs.



Convolutional Spatial Transformer (CST)

- The method incorporates the spatial transformer network (STN) [1] into their network architecture to enable an end-to-end learning.
- With the scale and rotation estimated, each patch centered at the reference pixel is normalized, similar to SIFT.

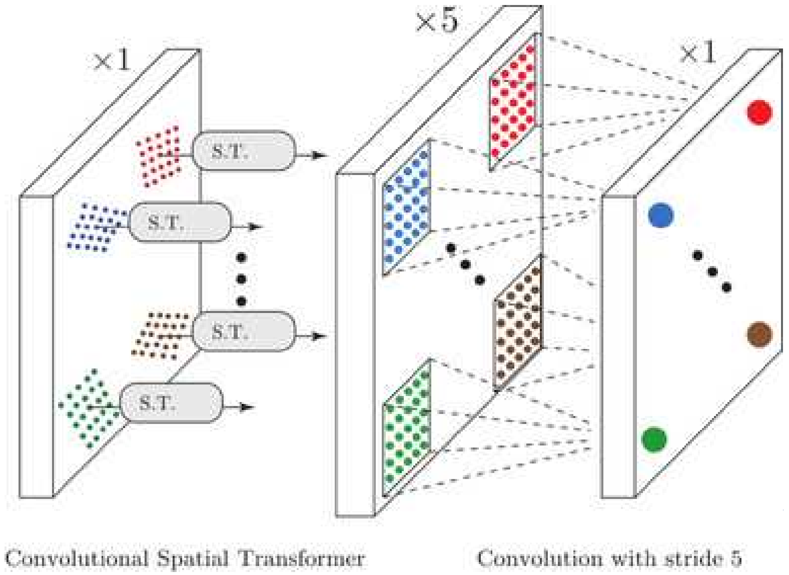

Scale
Rotation
$\theta \rightarrow \mathcal{T}_\theta$
Spatial Transformer
×1
×5
×1
S.T.
S.T.
S.T.
S.T.
Convolutional Spatial Transformer
Convolution with stride 5

(a) SIFT (b) Spatial transformer [1] (c) Convolutional spatial transformer

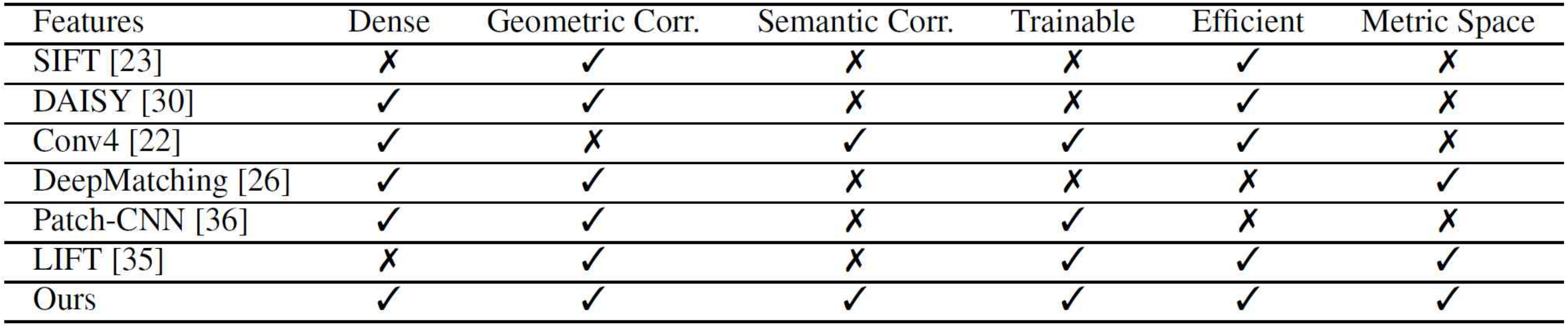
[1] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial Transformer Networks,” NIPS 2015

Convolutional Spatial Transformer (CST)

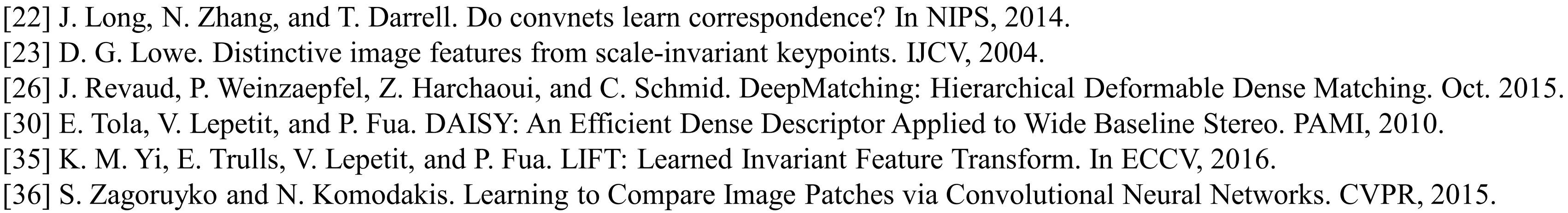
- ***Convolutional*** **Spatial Transformer**: Trick for addressing the geometry variance
  1) The CST takes an input from a lower layer and applies independent spatial transformation for each patch.
  2) The activations are normalized (transformed) independently, e.g., 5×5 window as below.
  3) The transformed activations are placed in a larger activation without overlap.
  4) Apply a successive convolution with the stride (Here, 5) to combine the transformed activations independently.



Comparison with Other Descriptors



Features	Dense	Geometric Corr.	Semantic Corr.	Trainable	Efficient	Metric Space
SIFT [23]	✗	✓	✗	✗	✓	✗
DAISY [30]	✓	✓	✗	✗	✓	✗
Conv4 [22]	✓	✗	✓	✓	✓	✗
DeepMatching [26]	✓	✓	✗	✗	✗	✓
Patch-CNN [36]	✓	✓	✗	✓	✗	✗
LIFT [35]	✗	✓	✗	✓	✓	✓
Ours	✓	✓	✓	✓	✓	✓



[22] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In NIPS, 2014.
[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 2004.
[26] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. DeepMatching: Hierarchical Deformable Dense Matching. Oct. 2015.
[30] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. PAMI, 2010.
[35] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. In ECCV, 2016.
[36] S. Zagoruyko and N. Komodakis. Learning to Compare Image Patches via Convolutional Neural Networks. CVPR, 2015.

UCN: Experimental Setup

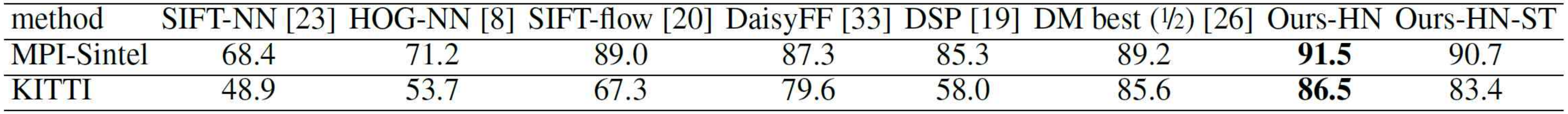
- **Performance measure: PCK@T**
  - The percentage of correct keypoints (PCK) metric with threshold T

- **Dataset**
  1. Geometric correspondence: KITTI 2015 Flow benchmark, MPI Sintel dataset
  2. Semantic correspondence: PASCAL-Berkeley dataset with keypoint annotations and a subset used by FlowWeb, Caltech-UCSD Bird dataset
  3. Camera motion estimation: raw KITTI driving sequences which include Velodyne scans, GPS and IMU measurements

Geometric Correspondence

- **Generating training data**
  - Randomly pick 1000 correspondences in KITTI, MPI Sintel image
  - Hard negative samples: a pair of correspondence when the nearest neighbor in the feature space is more than 16 pixels away from the ground truth correspondence

Matching performance PCK@10px on KITTI Flow 2015 and MPI-Sintel



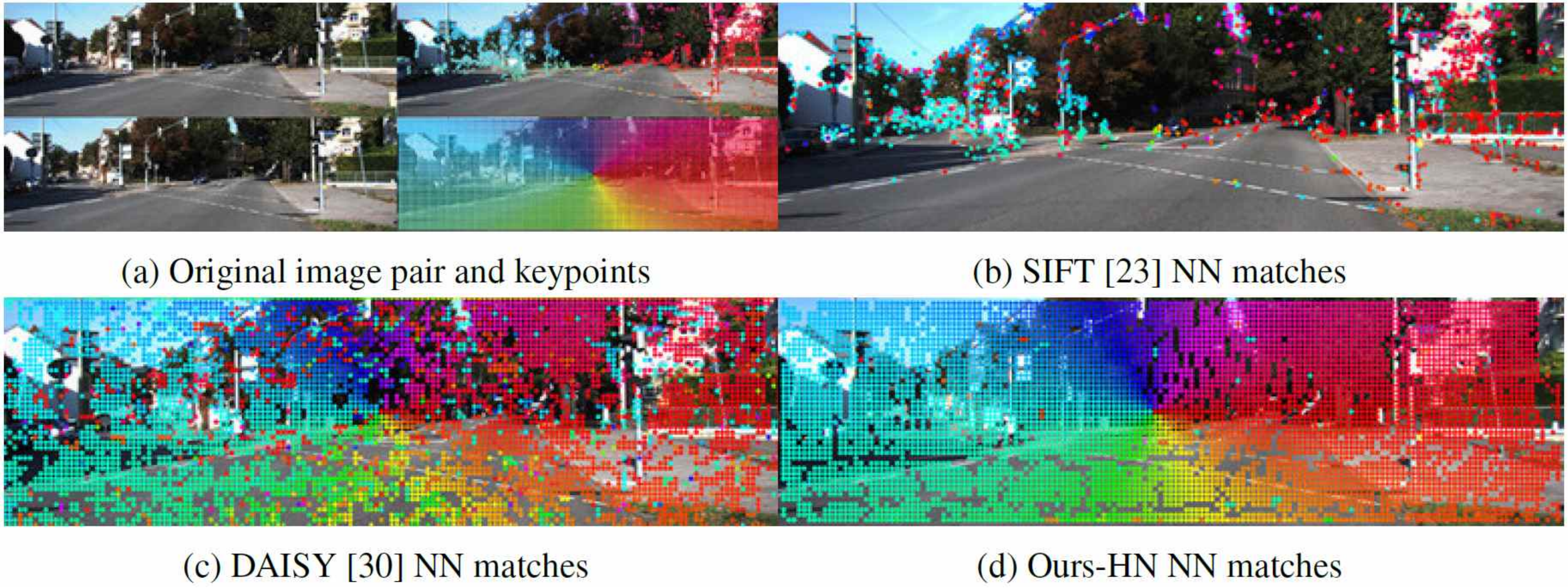
method	SIFT-NN [23]	HOG-NN [8]	SIFT-flow [20]	DaisyFF [33]	DSP [19]	DM best (½) [26]	Ours-HN	Ours-HN-ST
MPI-Sintel	68.4	71.2	89.0	87.3	85.3	89.2	**91.5**	90.7
KITTI	48.9	53.7	67.3	79.6	58.0	85.6	**86.5**	83.4

[20][26][33]: uses additional global optimization techniques, while UCN just employs WTA

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.
[19] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. CVPR 2013.
[20] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. PAMI, 33(5), May 2011.
[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 2004.
[26] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. DeepMatching: Hierarchical Deformable Dense Matching. 2015.
[33] H. Yang,W. Y. Lin, and J. Lu. DAISY filter flow: A generalized approach to discrete dense correspondences. In CVPR, 2014.

Geometric Correspondence

- Visualization of nearest neighbor (NN) matches on KITTI images



(a) Original image pair and keypoints

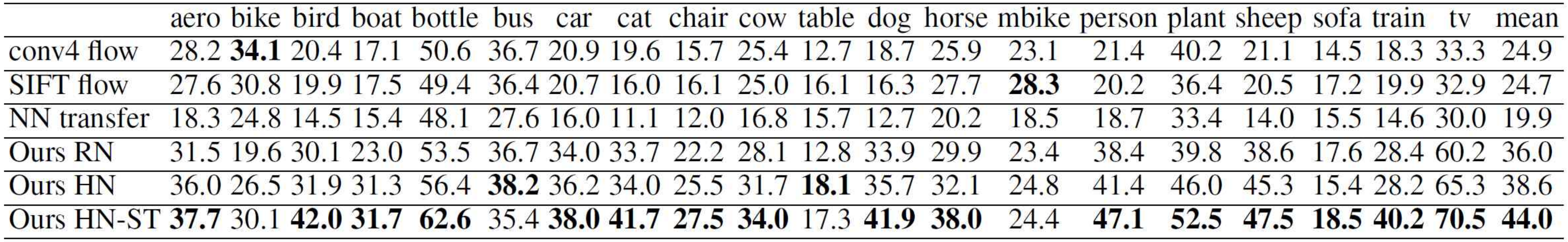
(b) SIFT [23] NN matches

(c) DAISY [30] NN matches

(d) Ours-HN NN matches

Semantic Correspondence

- **Per-class PCK on PASCAL-Berkeley correspondence dataset**



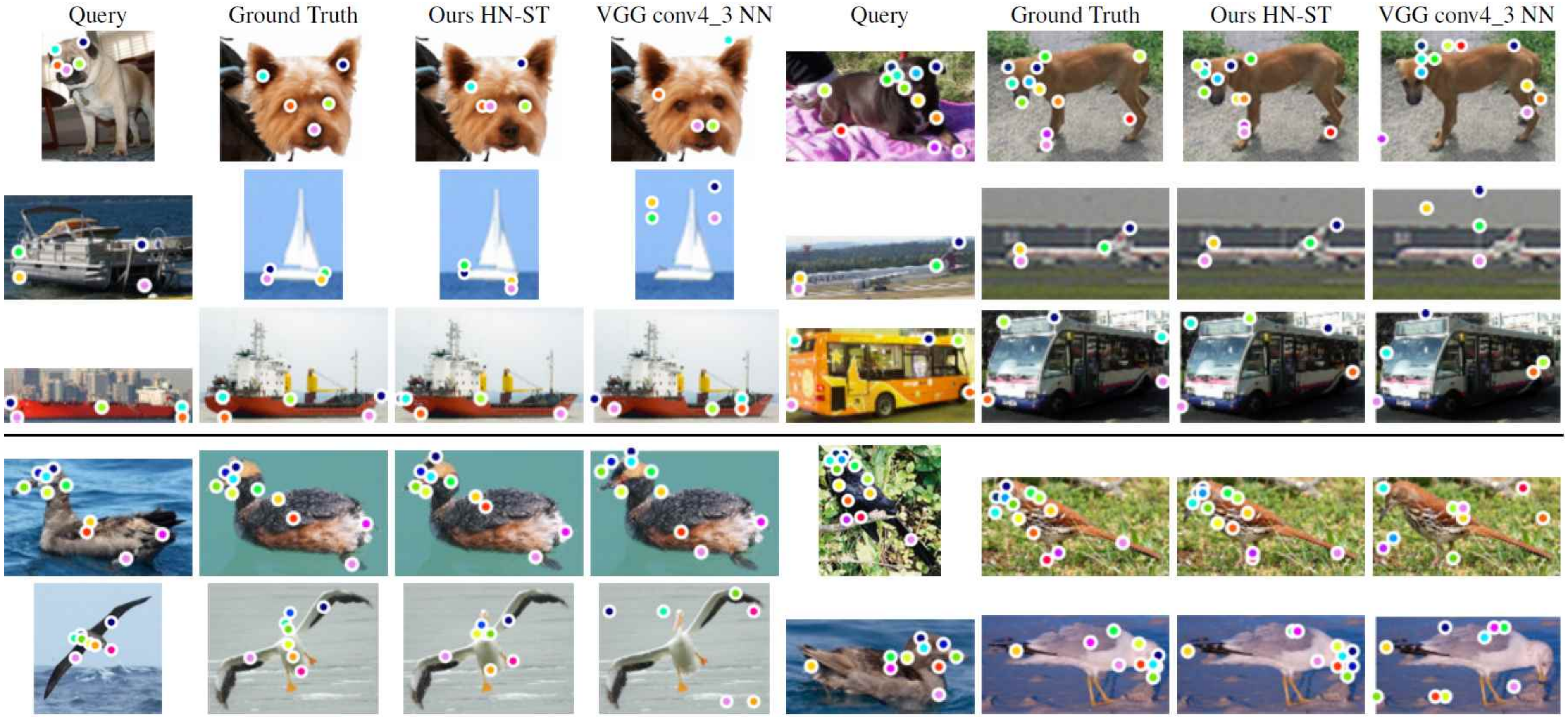
	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
conv4 flow	28.2	**34.1**	20.4	17.1	50.6	36.7	20.9	19.6	15.7	25.4	12.7	18.7	25.9	23.1	21.4	40.2	21.1	14.5	18.3	33.3	24.9
SIFT flow	27.6	30.8	19.9	17.5	49.4	36.4	20.7	16.0	16.1	25.0	16.1	16.3	27.7	**28.3**	20.2	36.4	20.5	17.2	19.9	32.9	24.7
NN transfer	18.3	24.8	14.5	15.4	48.1	27.6	16.0	11.1	12.0	16.8	15.7	12.7	20.2	18.5	18.7	33.4	14.0	15.5	14.6	30.0	19.9
Ours RN	31.5	19.6	30.1	23.0	53.5	36.7	34.0	33.7	22.2	28.1	12.8	33.9	29.9	23.4	38.4	39.8	38.6	17.6	28.4	60.2	36.0
Ours HN	36.0	26.5	31.9	31.3	56.4	**38.2**	36.2	34.0	25.5	31.7	**18.1**	35.7	32.1	24.8	41.4	46.0	45.3	15.4	28.2	65.3	38.6
Ours HN-ST	**37.7**	30.1	**42.0**	**31.7**	**62.6**	35.4	**38.0**	**41.7**	**27.5**	**34.0**	17.3	**41.9**	**38.0**	24.4	**47.1**	**52.5**	**47.5**	**18.5**	**40.2**	**70.5**	**44.0**

- Using PCK with αL(L: image size $\max(w, h)$, $\alpha = 0.1$)

- Ours-HN-ST: hard negative mining and spatial transformer
- Ours-HN: without spatial transformer
- Ours-RN: without spatial transformer and hard negative mining Instead, providing random negative samples that are at least certain pixels apart from the ground truth correspondence location instead

UCN: Experimental Results

- **Qualitative semantic correspondence results**
  - PASCAL-Berkeley keypoint annotation and Caltech-UCSD Bird dataset



Conclusion of UCN

- **Key contributions**
  - Correspondence contrastive loss in a fully convolutional manner
  - On-the-fly hard negative mining
  - Convolutional spatial transformer network

→ More efficient training, accurate gradient computations, faster testing and local patch normalization

→ Outperform prior state-of-the-art on geometric and semantic correspondence tasks, even without using any spatial priors or global optimization

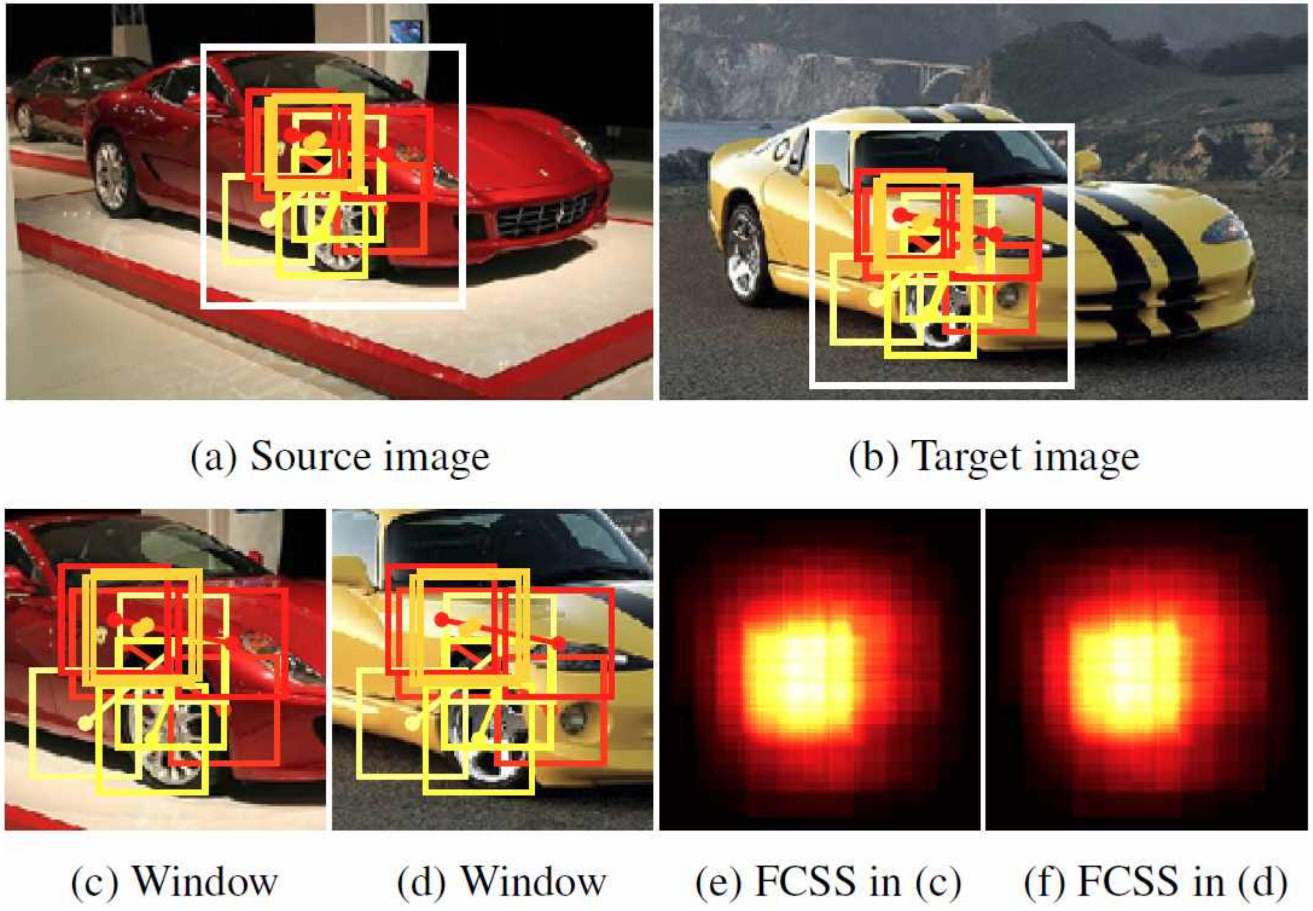
Remaining Challenges

- **Hand-crafted feature descriptors**
  - Finding a way of handling affine transform or projective transform
  - More generic framework for dealing with photometric distortion

- **Learning based descriptors**
  - Addressing both geometric and photometric variations in an end-to-end manner in ConvNet
  - Trade-off between Speed vs. Geometric Invariance
  - Hybrid approaches benefiting from a plenty of hand-crafted feature descriptors, when dealing with geometric and photometric variations

One More Thing…

- Ongoing work along semantic descriptors



(a) Source image

(b) Target image

(c) Window

(d) Window

(e) FCSS in (c)

(f) FCSS in (d)

Ongoing work along semantic descriptors

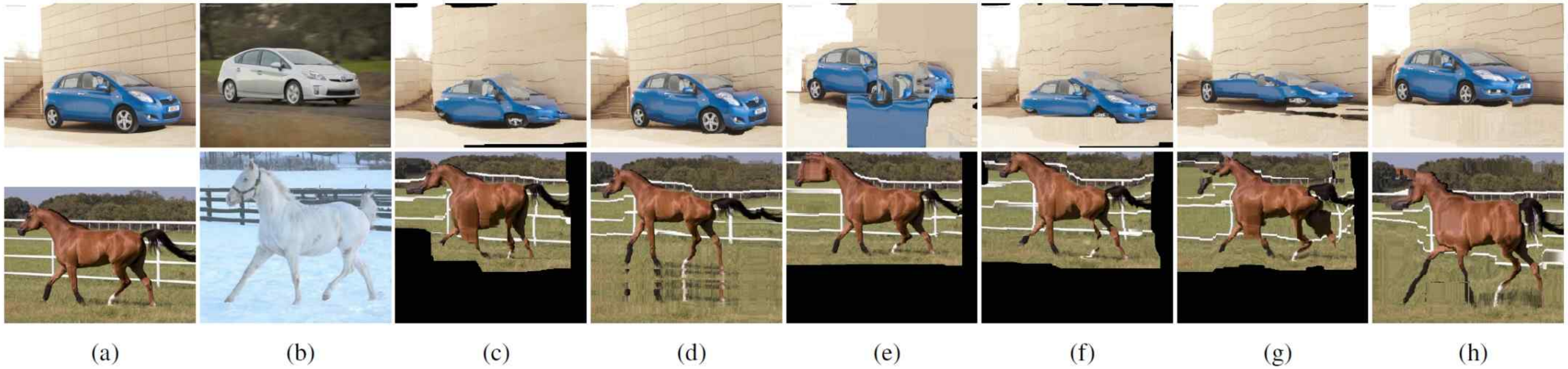


Figure 6. Qualitative results on the Taniai benchmark [45]: (a) source image, (b) target image, (c) SIFT [34], (d) DASC [25], (e) DeepD. [41], (f) MatchN. [19], (g) VGG [42], and (h) FCSS. The source images were warped to the target images using correspondences.

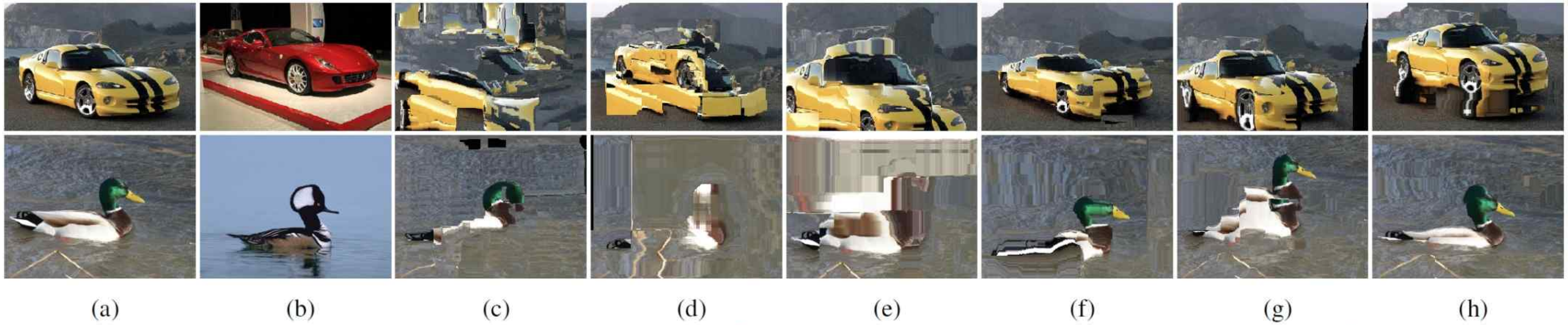


Figure 7. Qualitative results on the Proposal Flow benchmark [18]: (a) source image, (b) target image, (c) DAISY [46], (d) DeepD. [41], (e) DeepC. [51], (f) LIFT [50], (g) VGG [42], and (h) FCSS. The source images were warped to the target images using correspondences.