Siamese/Triplet Networks

Jia-Bin Huang

Virginia Tech

ECE 6554 Advanced Computer Vision

Today's class

- Discussions
- Review important concepts of visual recognition
  - Instance recognition
  - Category recognition
  - Supervised pre-training
  - Understanding/visualizing
  - Segmentation networks
- Siamese/Triplet networks for metric learning

Discussion – Think-pair-share

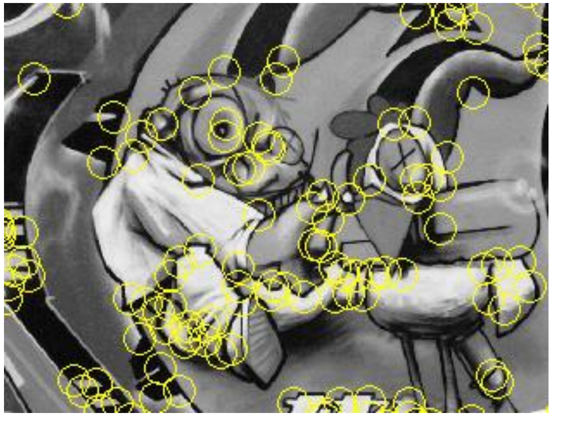
- FaceNet: A unified embedding for face recognition and clustering. F. Schroff, D. Kalenichenko, J Philbin, ICCV 2015

- Discuss
  - strength,
  - weakness, and
  - potential extension

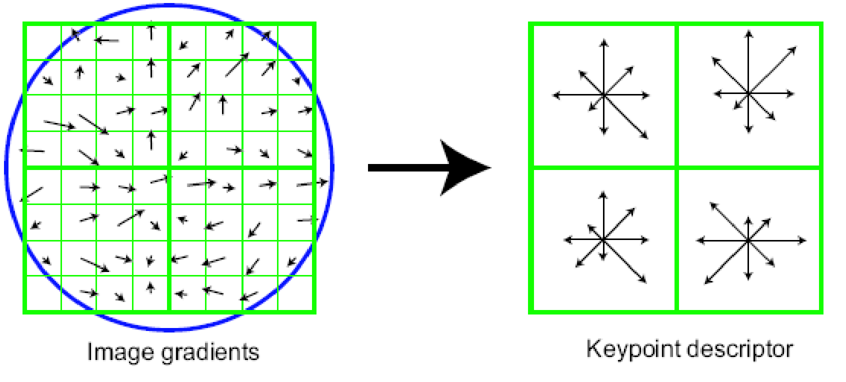
- Share with class

Keypoint detection and descriptors

- **Keypoint detection**: repeatable and distinctive
  - Corners, blobs, stable regions
  - Harris, DoG



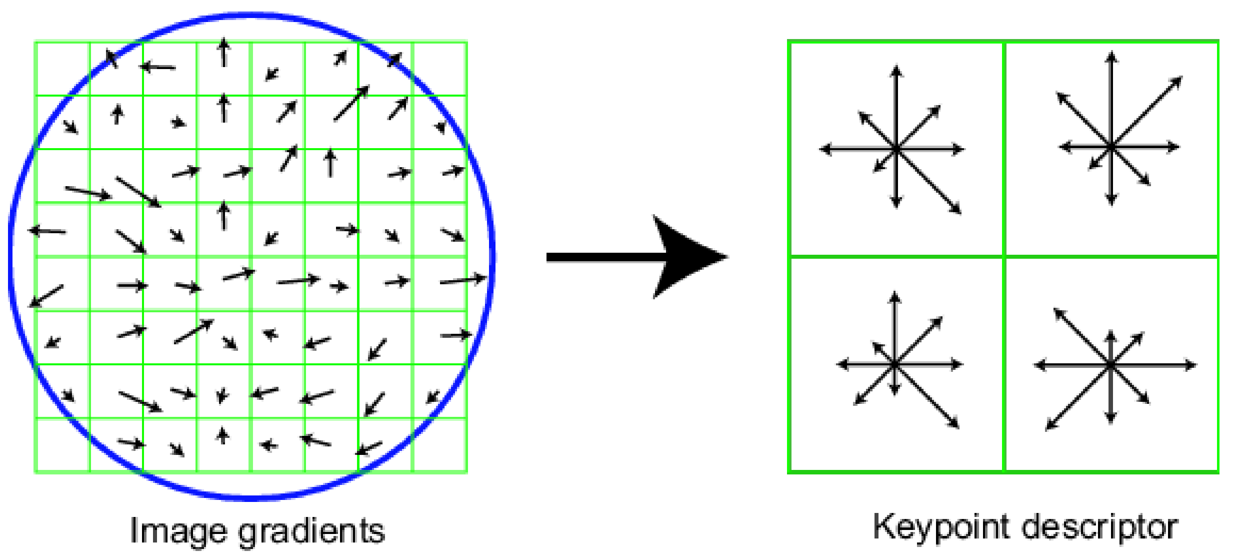
- **Descriptors**: robust and selective
  - spatial histograms of orientation
  - SIFT



SIFT descriptor

Full version

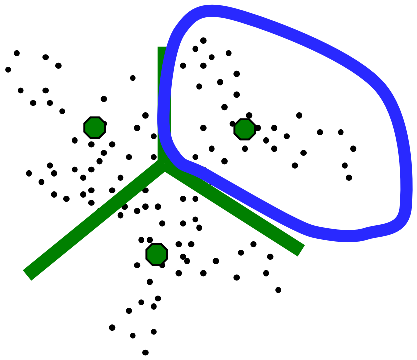
- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



Adapted from slide by David Lowe

Visual Words

- Example: each group of patches belongs to the same visual word



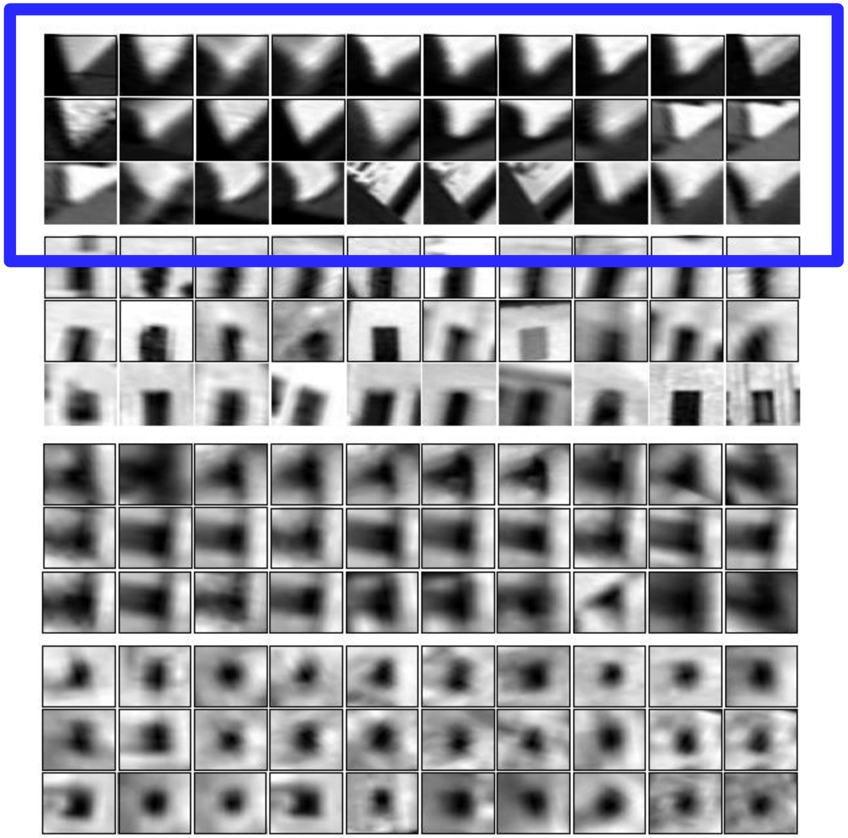
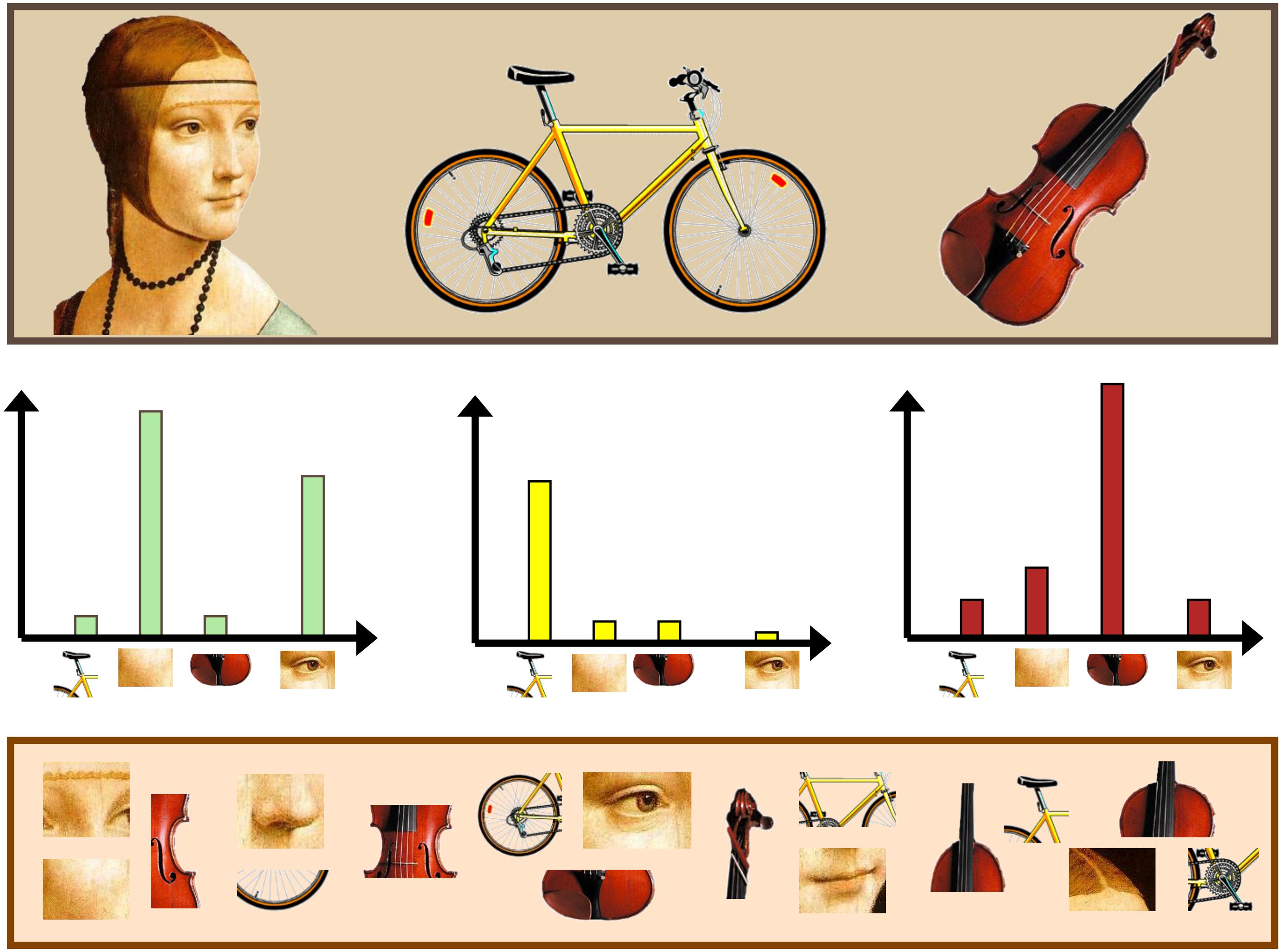


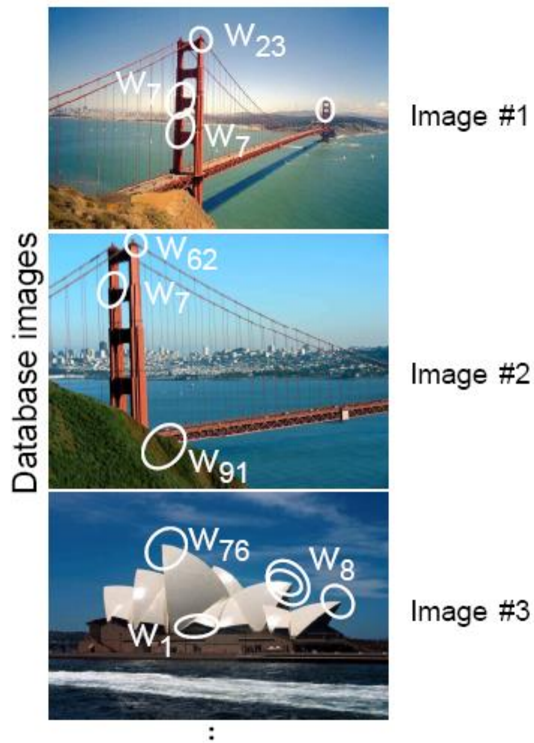
Figure from Sivic & Zisserman, ICCV 2003

Kristen Grauman

Bag of Words Models



Inverted file index



Word #	Image #
1	3
2 …	
7	1, 2
8	3
9	
10 …	
91	2
⋮	⋮

…

- Database images are loaded into the index mapping words to image numbers

Kristen Grauman

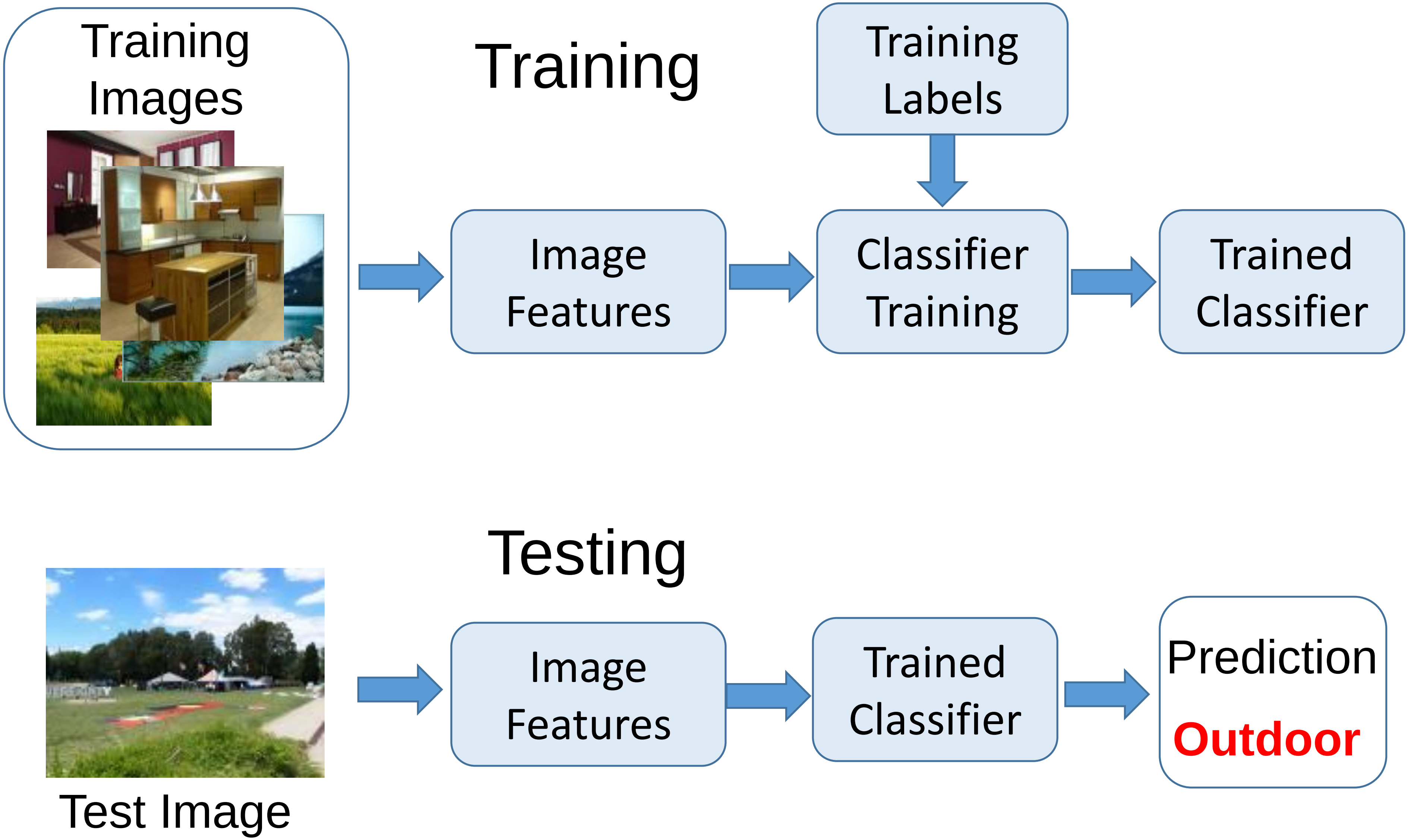
Spatial Verification: two basic strategies

- **RANSAC**
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible transformations
    - e.g., “success” if find a transformation with > N inlier correspondences

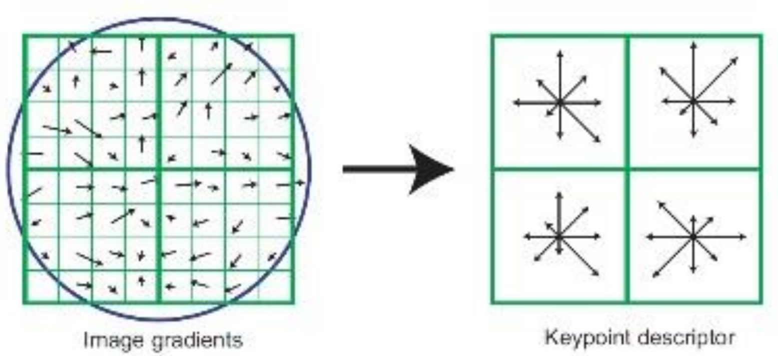
- **Generalized Hough Transform**
  - Let each matched feature cast a vote on location, scale, orientation of the model object
  - Verify parameters with enough votes

Kristen Grauman

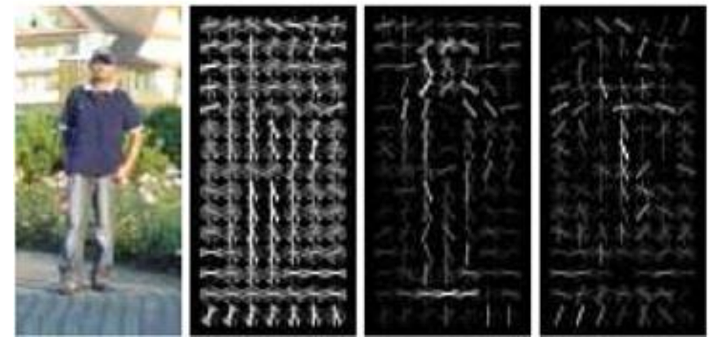
Image Categorization



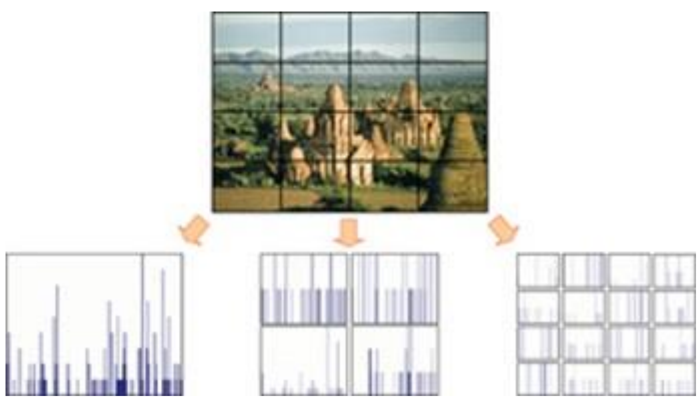
Features are the Keys



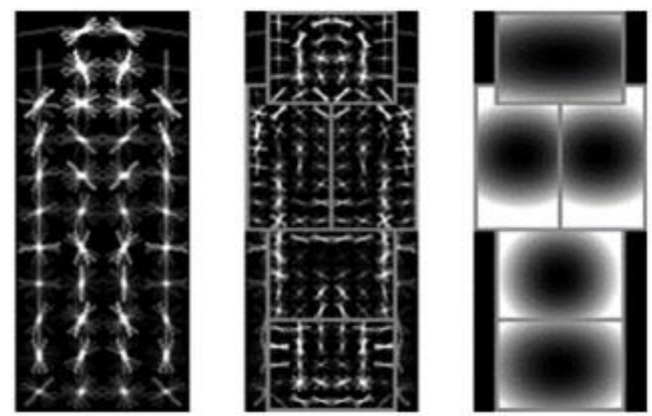
SIFT [Loewe IJCV 04]



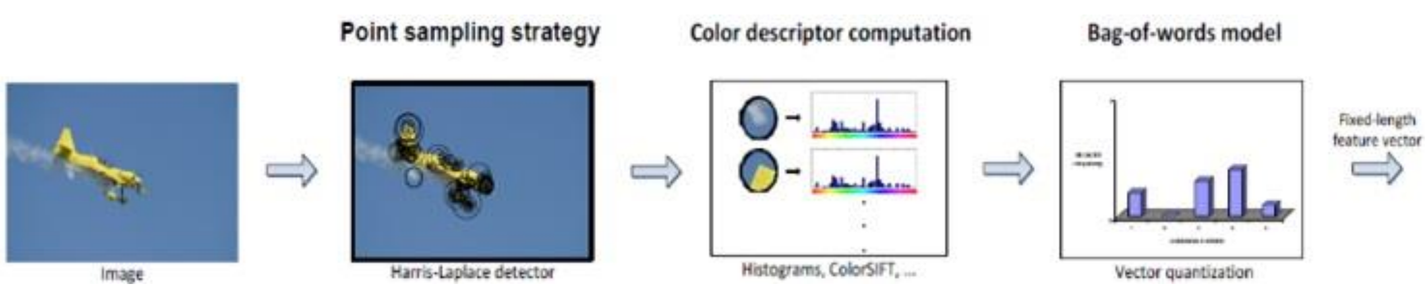
HOG [Dalal and Triggs CVPR 05]



SPM [Lazebnik et al. CVPR 06]



DPM [Felzenszwalb et al. PAMI 10]



Color Descriptor [Van De Sande et al. PAMI 10]

Putting them together

- Local connectivity
- Weight sharing
- Handling multiple input channels
- Handling multiple output maps

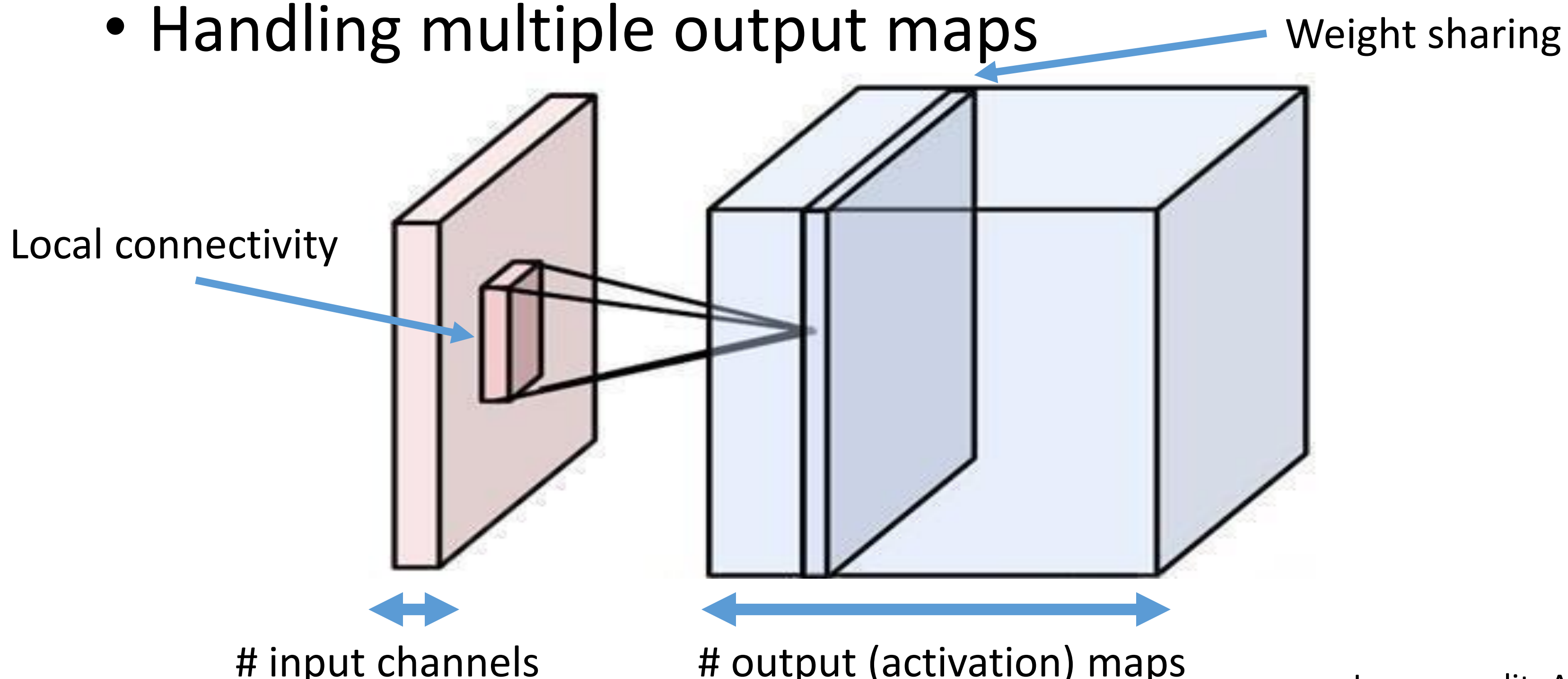
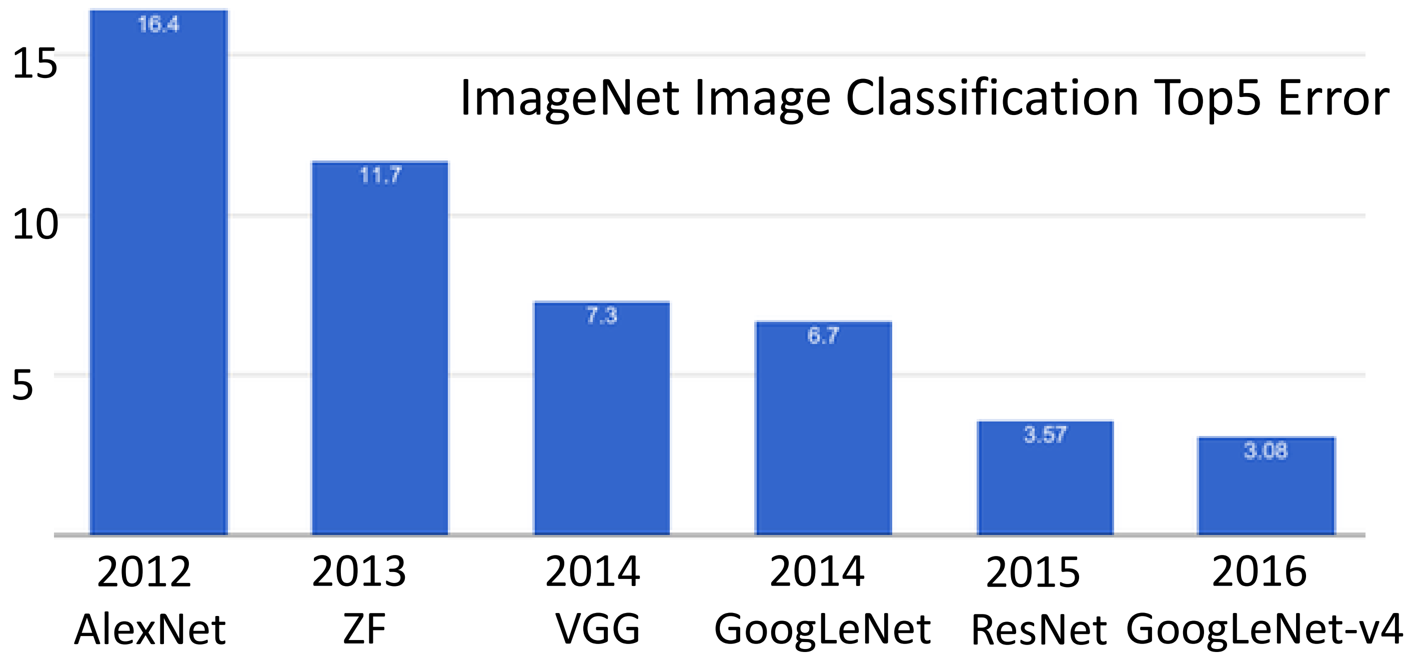


Image credit: A. Karpathy

Progress on ImageNet



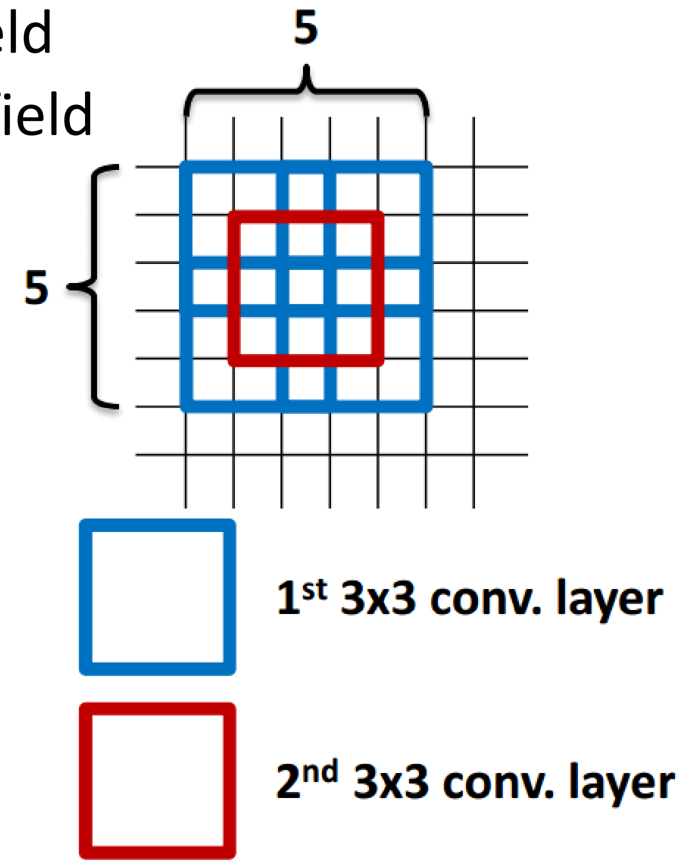
VGG-Net

- The deeper, the better

- Key design choices:
    - 3x3 conv. Kernels
      - very small
    - conv. stride 1
      - no loss of information
- Other details:
    - Rectification (ReLU) non-linearity
    - 5 max-pool layers (x2 reduction)
    - no normalization
    - 3 fully-connected (FC) layers

VGG-Net

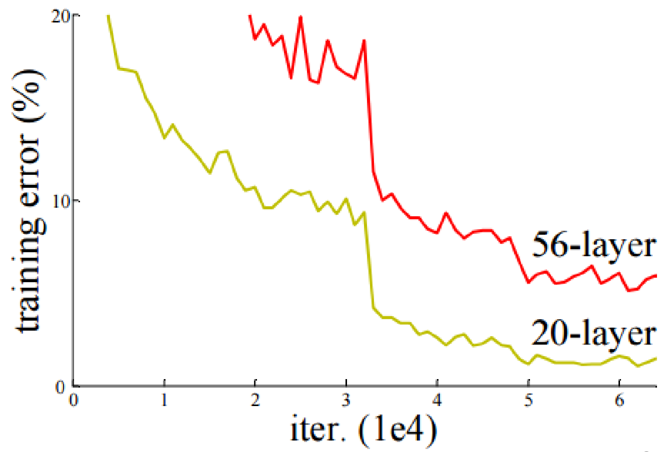
- Why 3x3 layers?
  - Stacked conv. layers have a large receptive field
  - two 3x3 layers – 5x5 receptive field
  - three 3x3 layers – 7x7 receptive field

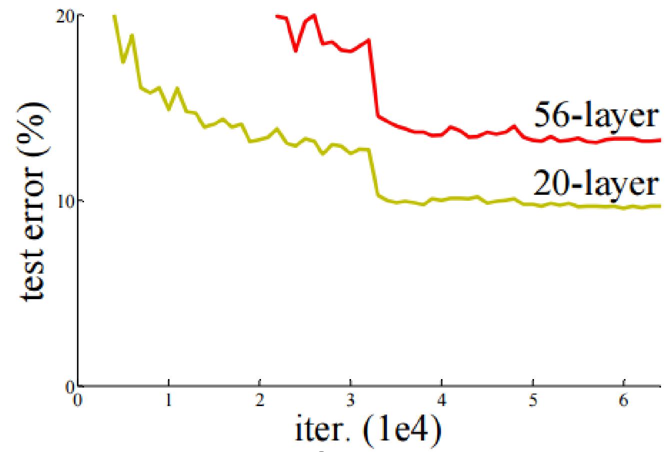


- More non-linearity
  - Less parameters to learn
  - ~140M per net

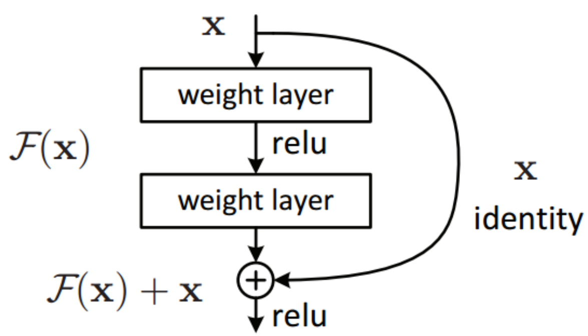
ResNet

- Can we just increase the #layer?





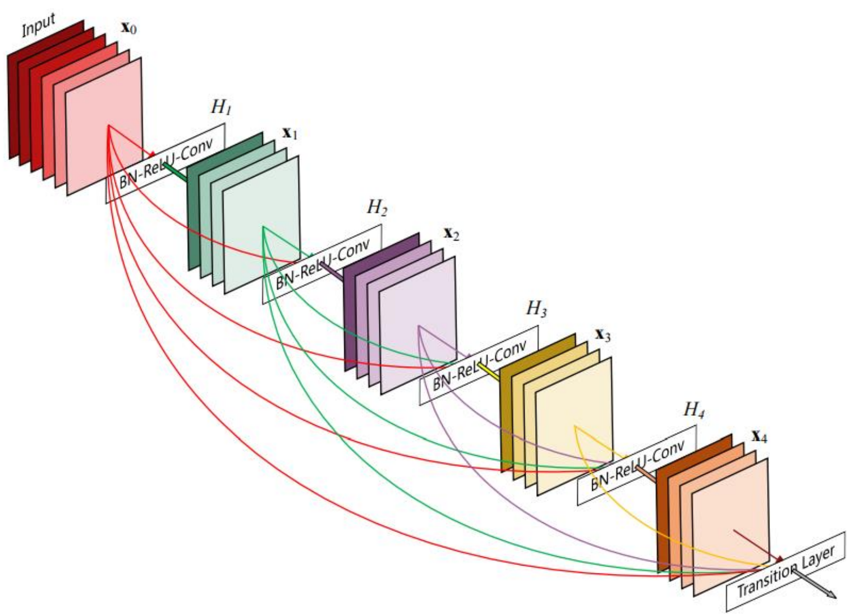
- How can we train very deep network?
  - Residual learning

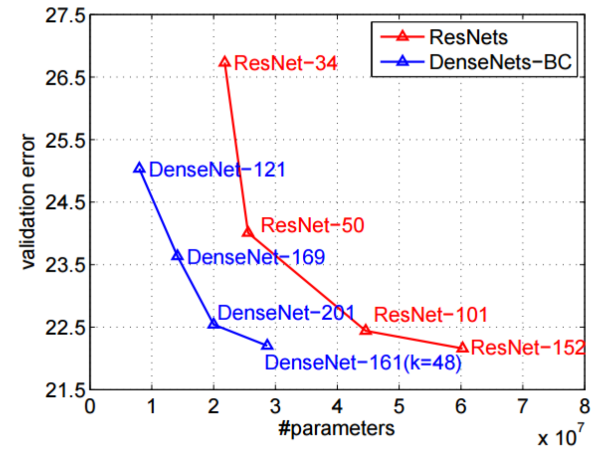


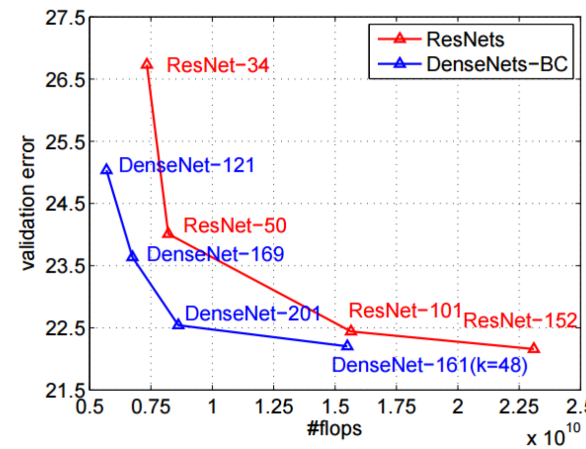
method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

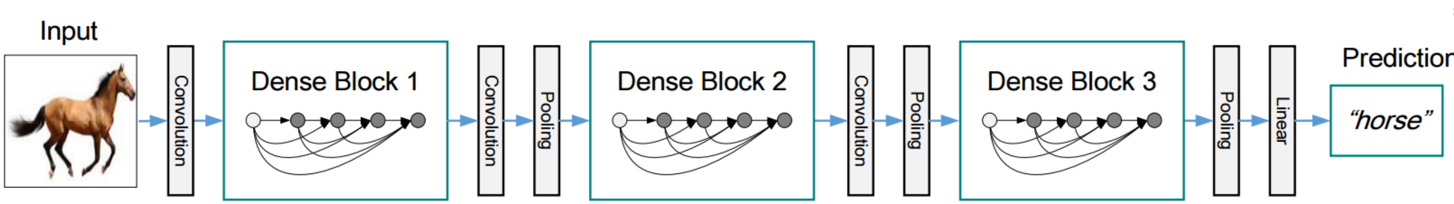
DenseNet

- Shorter connections (like ResNet) help
- Why not just connect them all?



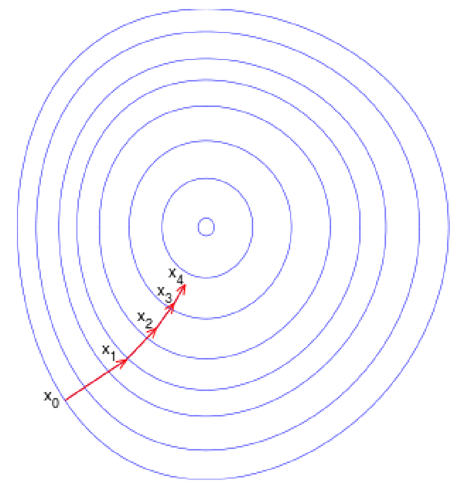






Training CNN with gradient descent

- A CNN as composition of functions
$$f_{\boldsymbol{w}}(\boldsymbol{x}) = f_L(\ldots\ (f_2(f_1(\boldsymbol{x};\boldsymbol{w}_1);\boldsymbol{w}_2)\ldots;\boldsymbol{w}_L)$$
- Parameters
$$\boldsymbol{w} = (\boldsymbol{w}_{\mathbf{1}},\boldsymbol{w}_{\mathbf{2}},\ldots\boldsymbol{w}_L)$$
- Empirical loss function
$$L(\boldsymbol{w}) = \frac{1}{n}\sum_i l(z_i, f_{\boldsymbol{w}}(\boldsymbol{x}_{\boldsymbol{i}}))$$



- Gradient descent

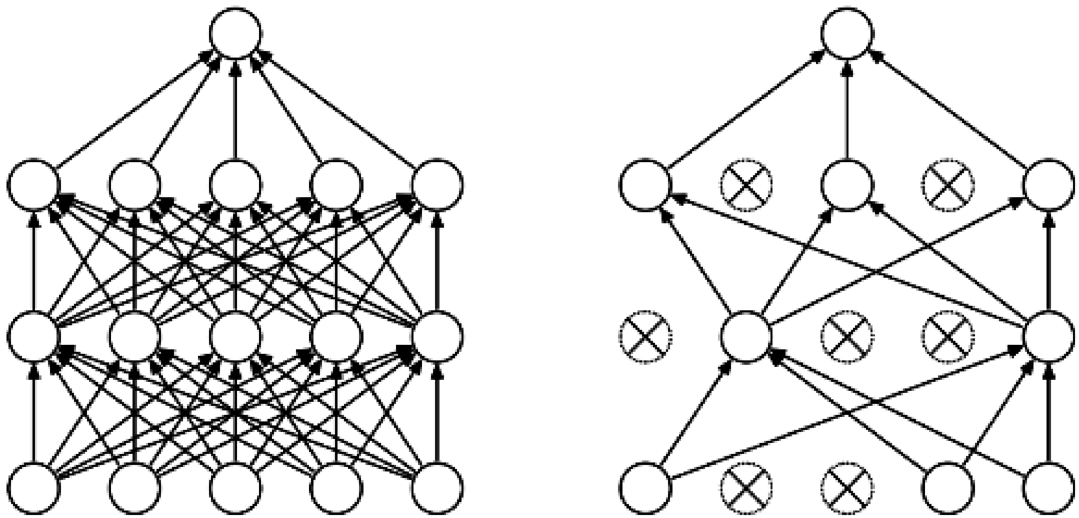
New weight
$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^{t} - \eta_t \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}}(\boldsymbol{w}^{t})$$

Old weight

Learning rate

Gradient

Dropout



(a) Standard Neural Net (b) After applying dropout.

Intuition: successful conspiracies

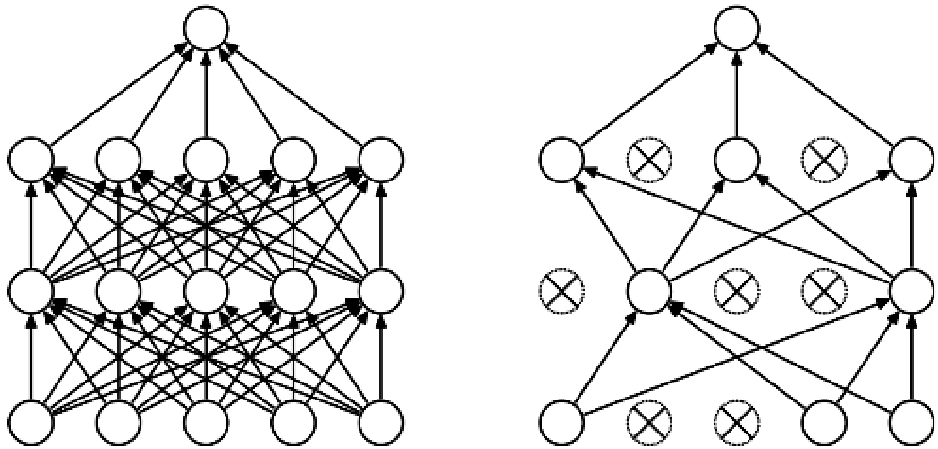
- 50 people planning a conspiracy

- Strategy A: plan a big conspiracy involving 50 people
    - Likely to fail. 50 people need to play their parts correctly.

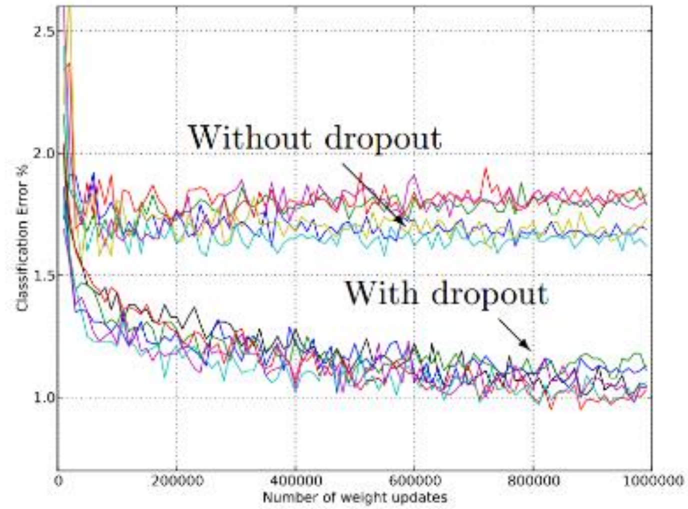
- Strategy B: plan 10 conspiracies each involving 5 people
    - Likely to succeed!

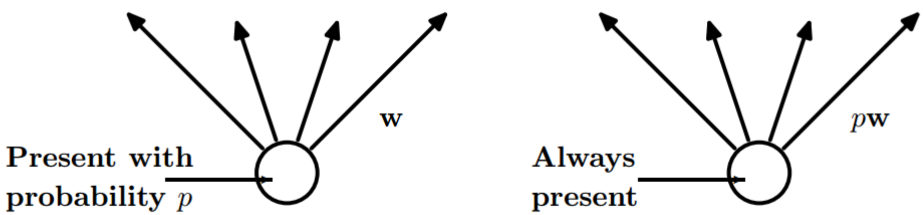
Dropout: A simple way to prevent neural networks from overfitting [Srivastava JMLR 2014]

Dropout



(a) Standard Neural Net (b) After applying dropout.





(a) At training time (b) At test time

Main Idea: approximately combining exponentially many different neural network architectures efficiently

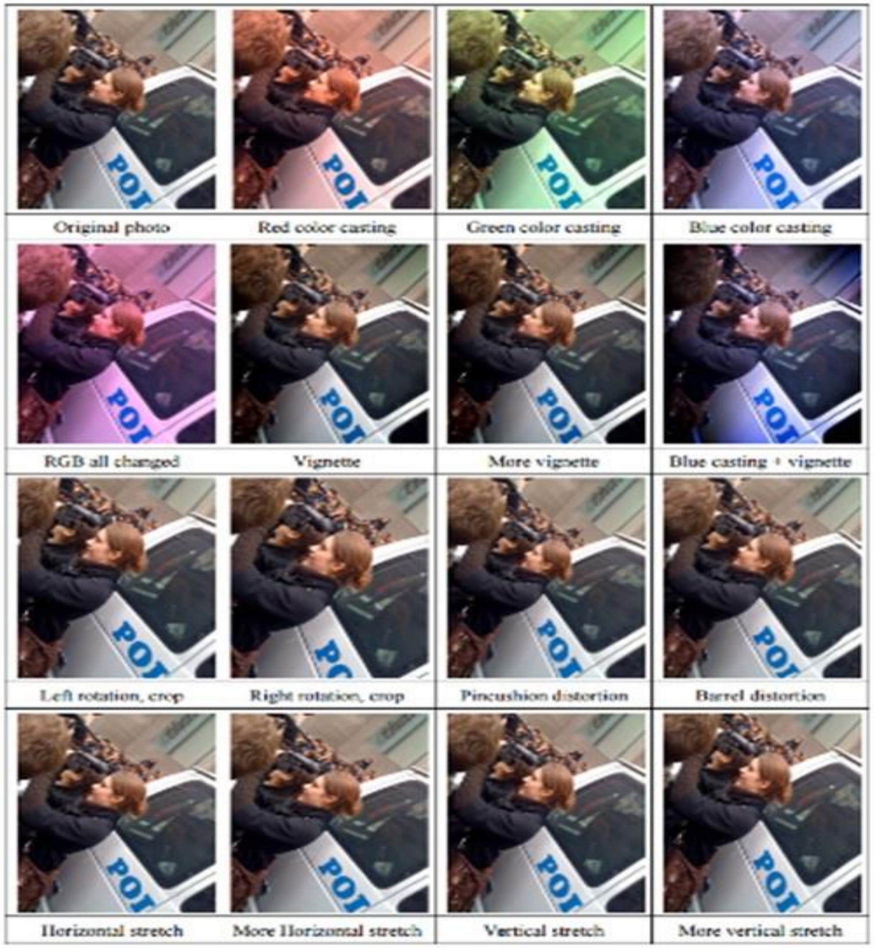
Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SVM on Fisher Vectors of Dense SIFT and Color Statistics	-	-	27.3
Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT	-	-	26.2
Conv Net + dropout (Krizhevsky et al., 2012)	40.7	18.2	-
Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012)	38.1	16.4	16.4

Table 6: Results on the ILSVRC-2012 validation/test set.

Dropout: A simple way to prevent neural networks from overfitting [Srivastava JMLR 2014]

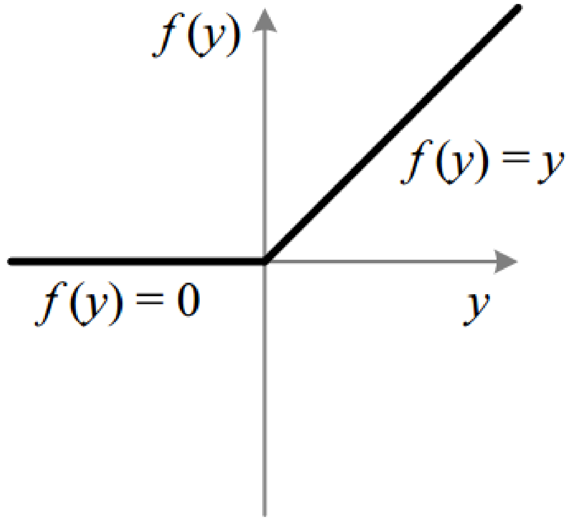
Data Augmentation (Jittering)

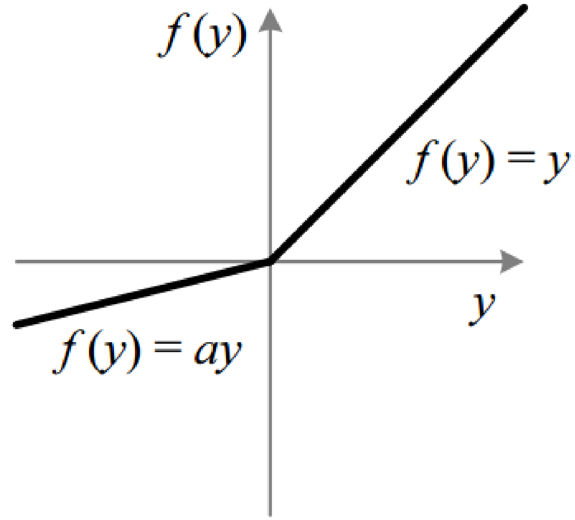
- Create *virtual* training samples
  - Horizontal flip
  - Random crop
  - Color casting
  - Geometric distortion



Deep Image [Wu et al. 2015]

Parametric Rectified Linear Unit





	team	top-5 (**test**)
in competition ILSVRC 14	MSRA, SPP-nets [11]	8.06
	VGG [25]	7.32
	GoogLeNet [29]	6.66
post-competition	VGG [25] (arXiv v5)	6.8
	Baidu [32]	5.98
	MSRA, PReLU-nets	**4.94**

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification [He et al. 2015]

Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

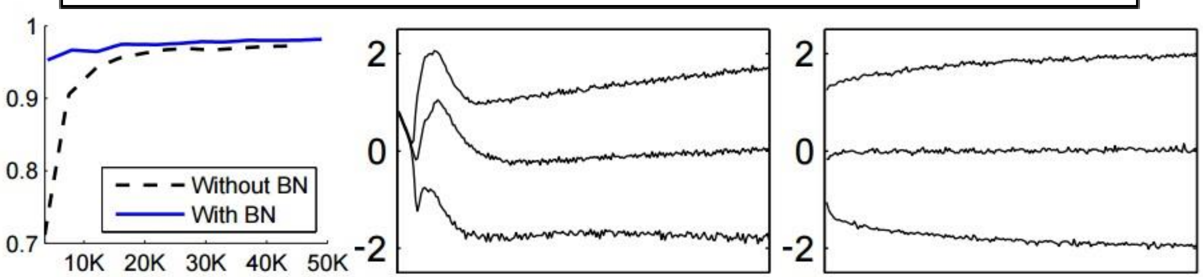
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_\mathcal{B})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}} \qquad \text{// normalize}$$

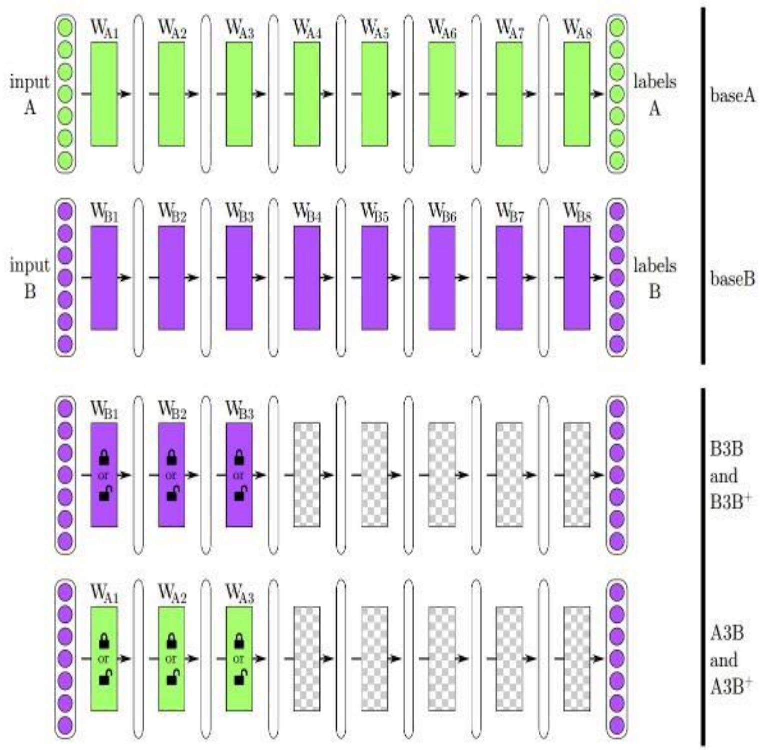
$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

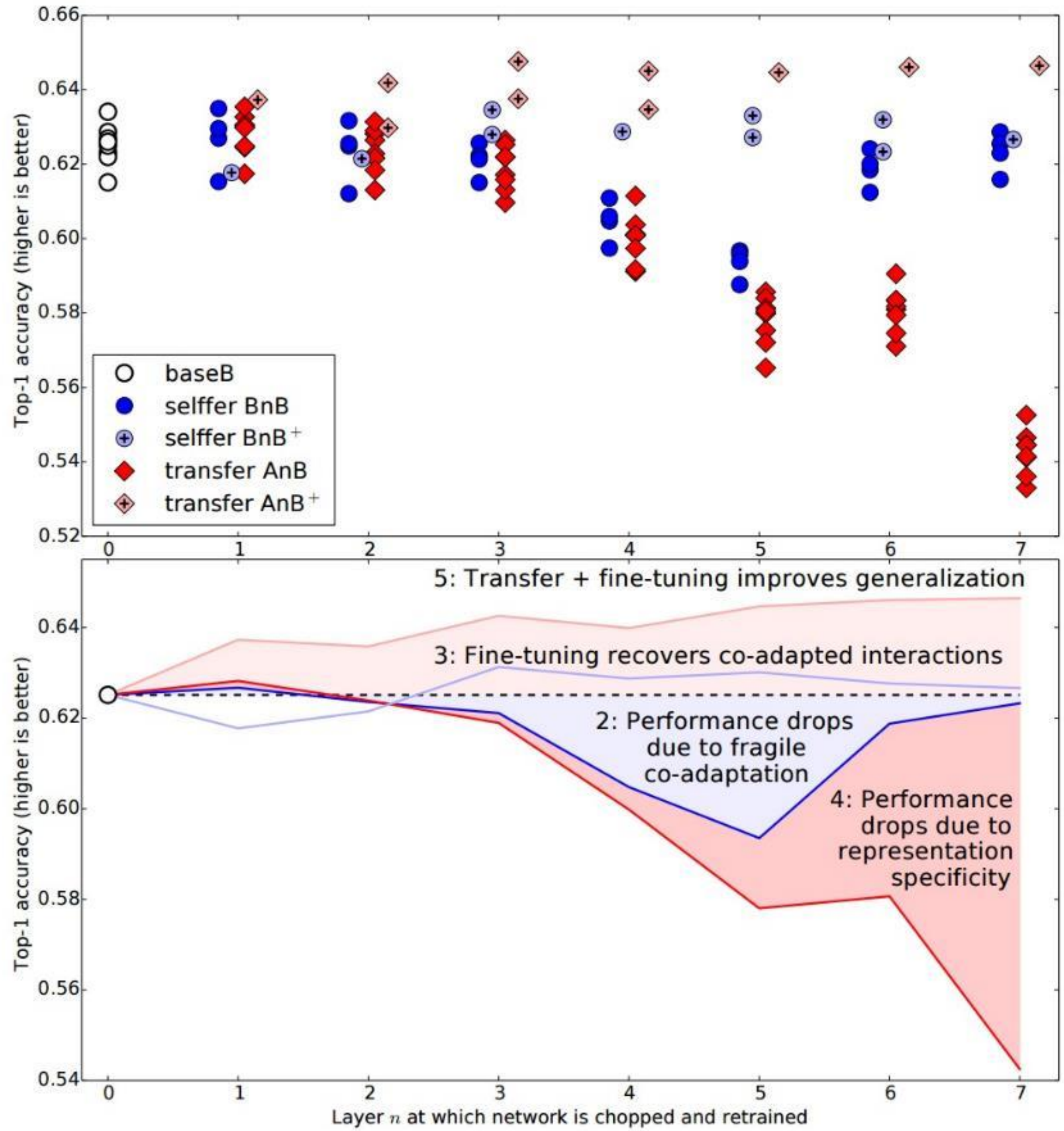


(a) (b) Without BN (c) With BN

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Ioffe and Szegedy 2015]

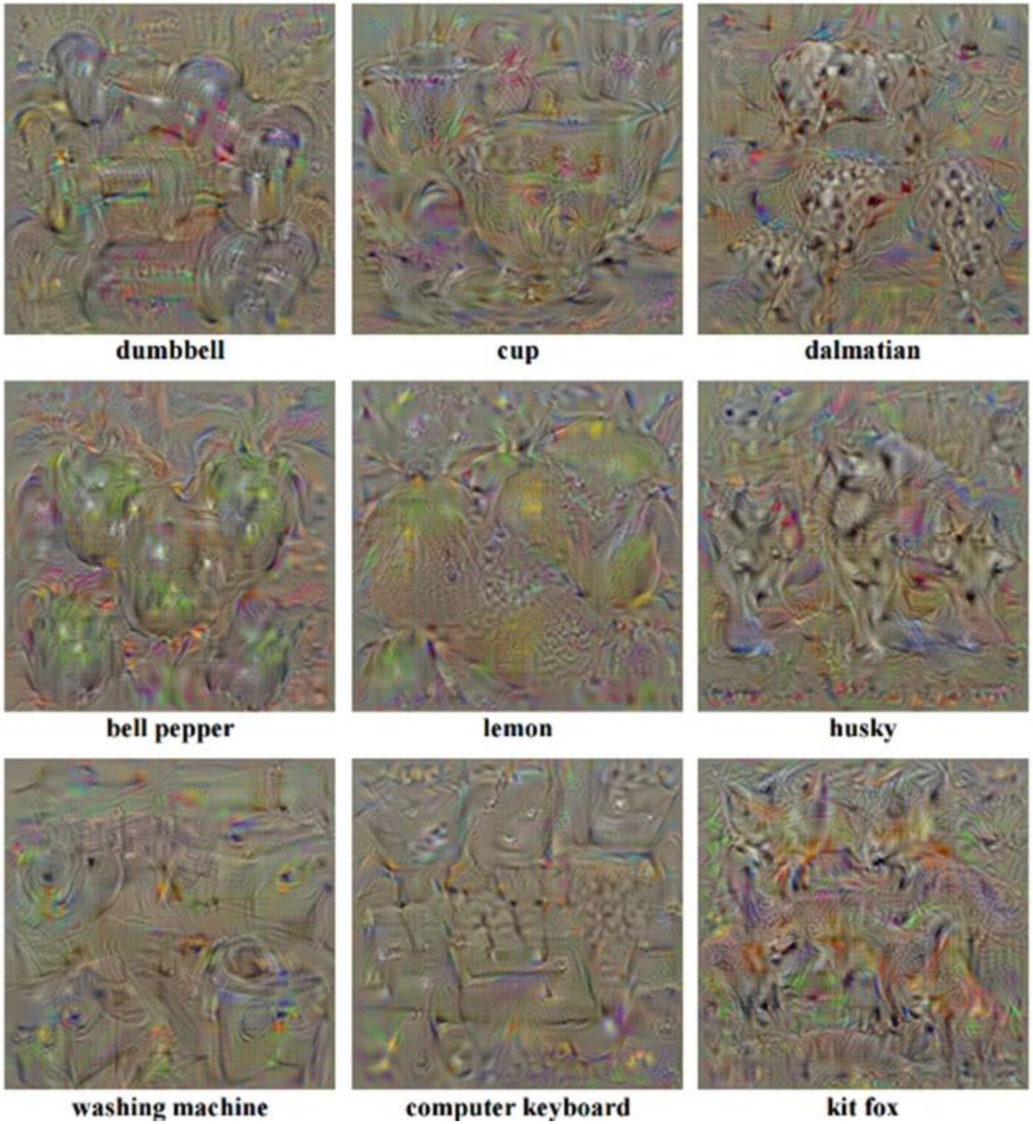
How transferable are features in CNN?





How transferable are features in deep neural networks [Yosinski NIPS 2014]

Find images that maximize some class scores

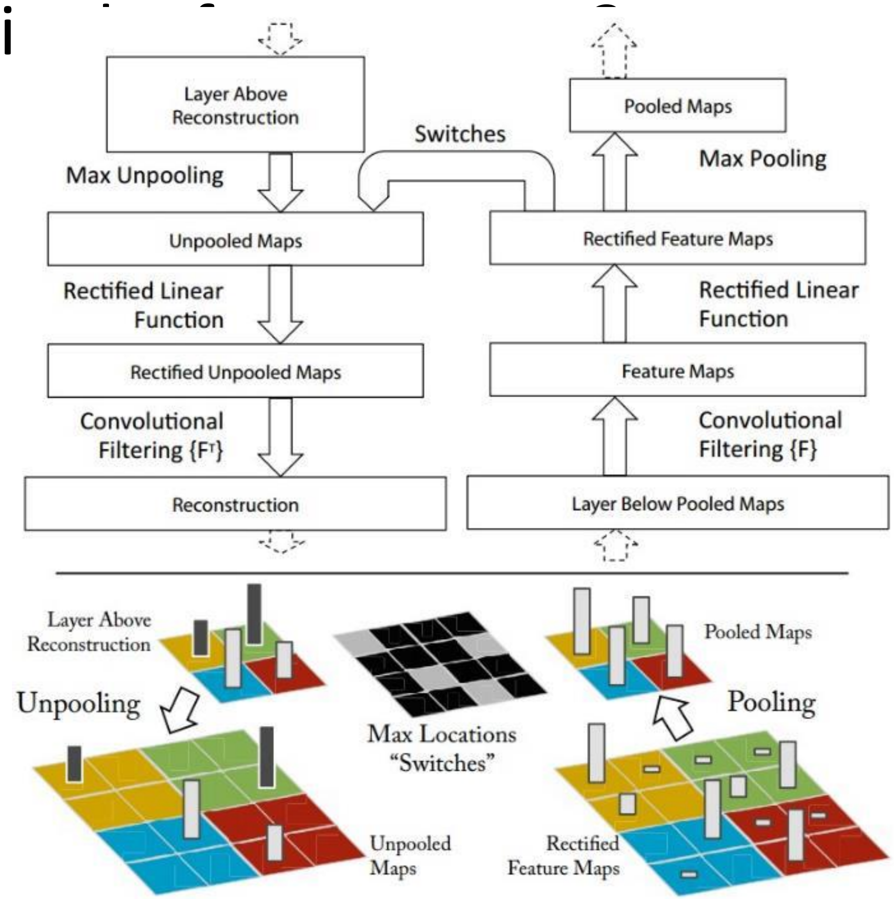


person: HOG template

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
[Simonyan et al. ICLR Workshop 2014]

Map activation back to the input pixel space

- What input pattern originally caused a given activation i

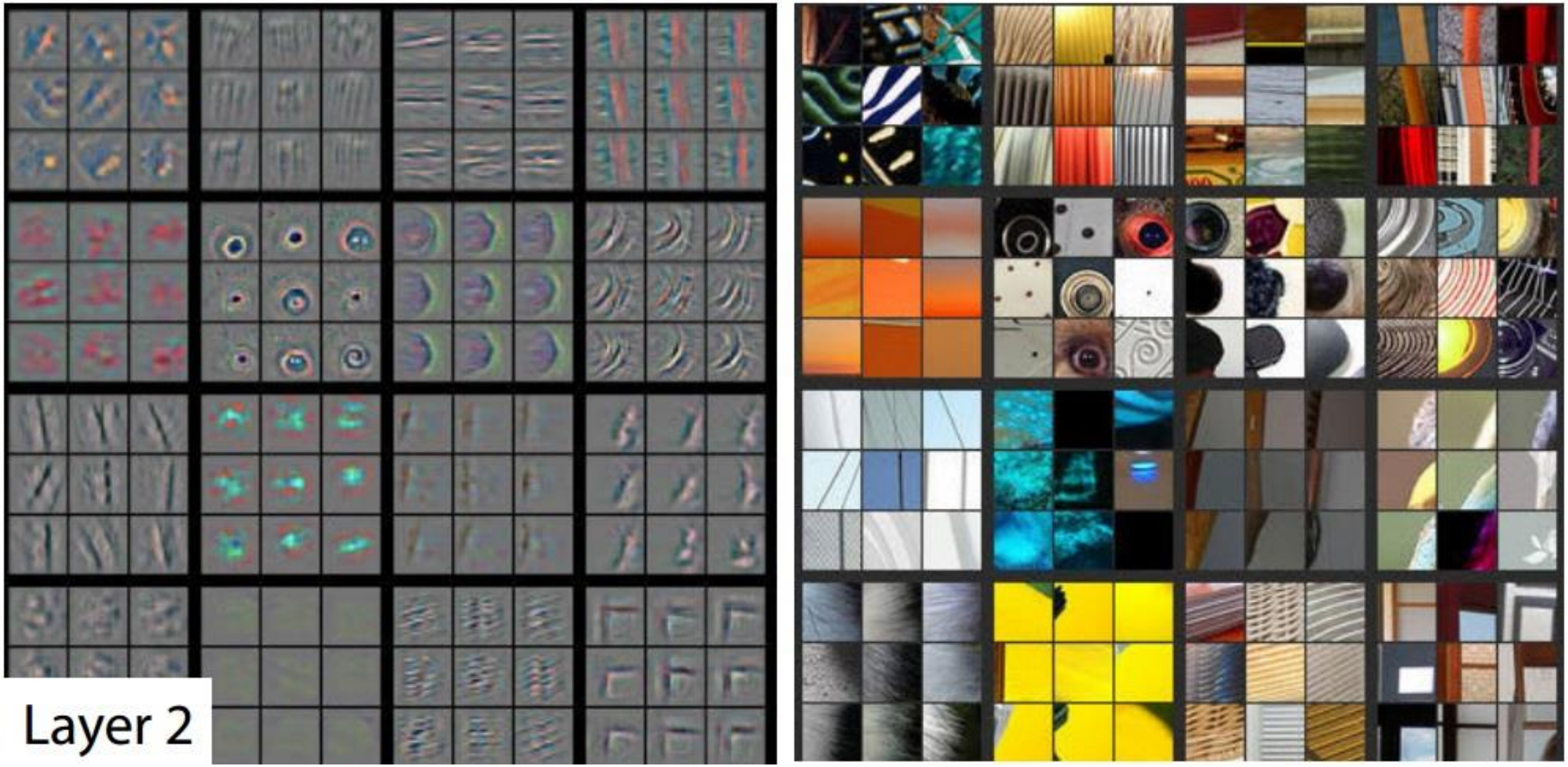


Visualizing and Understanding Convolutional Networks [[Zeiler and Fergus, ECCV 2014](#)]

Layer 1

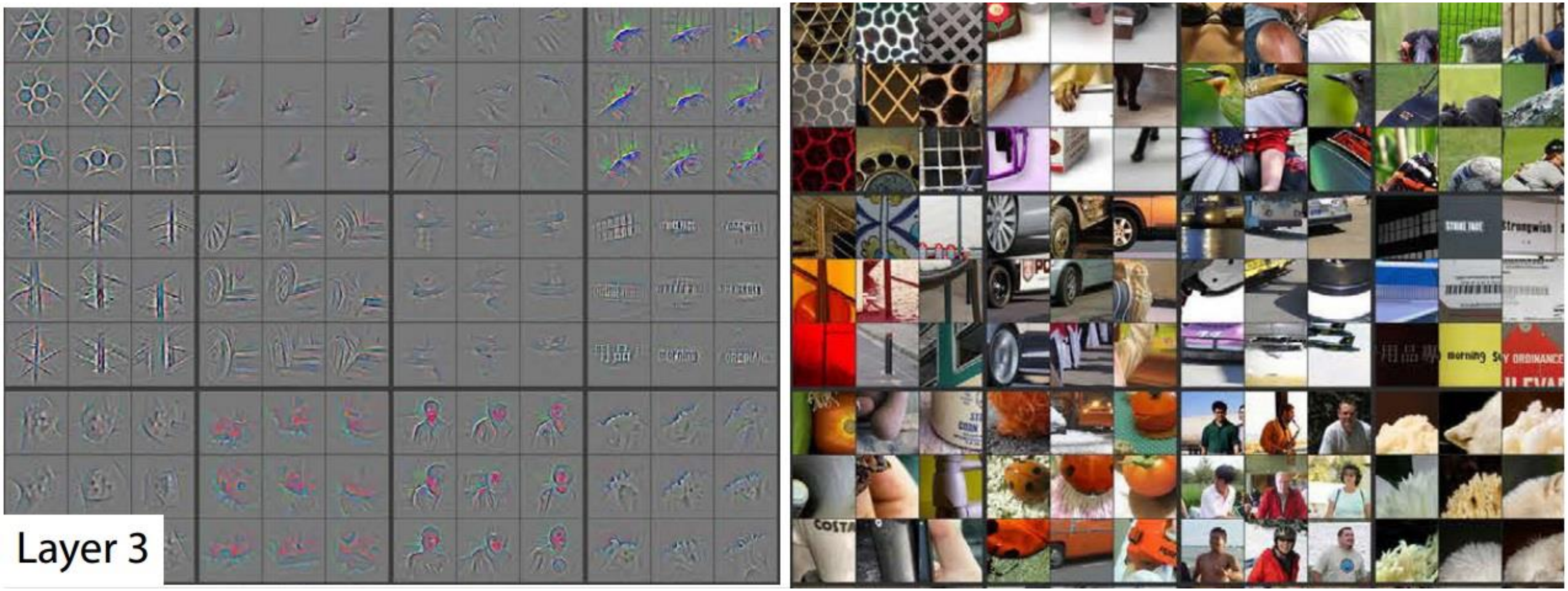
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 2



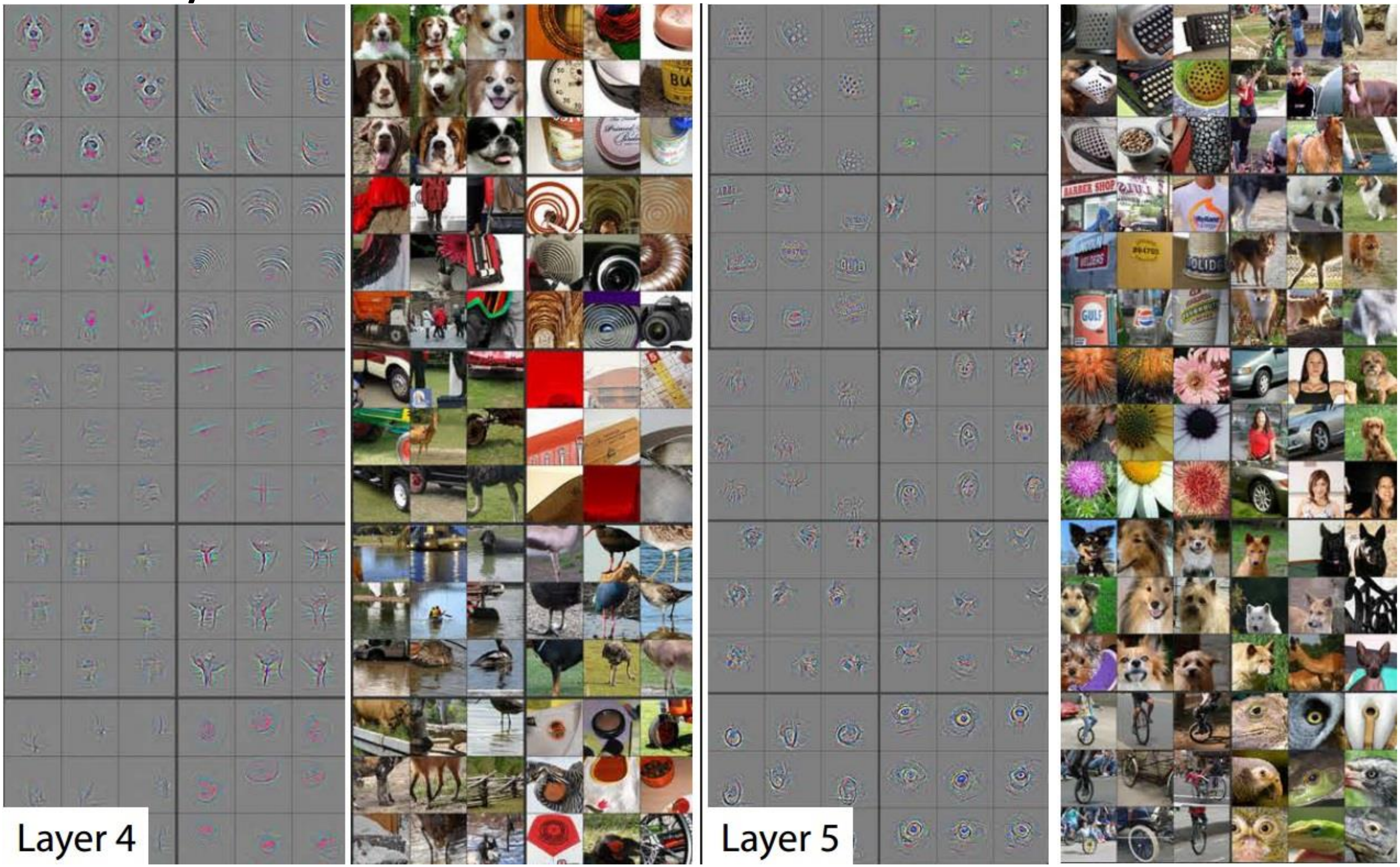
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 3



Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

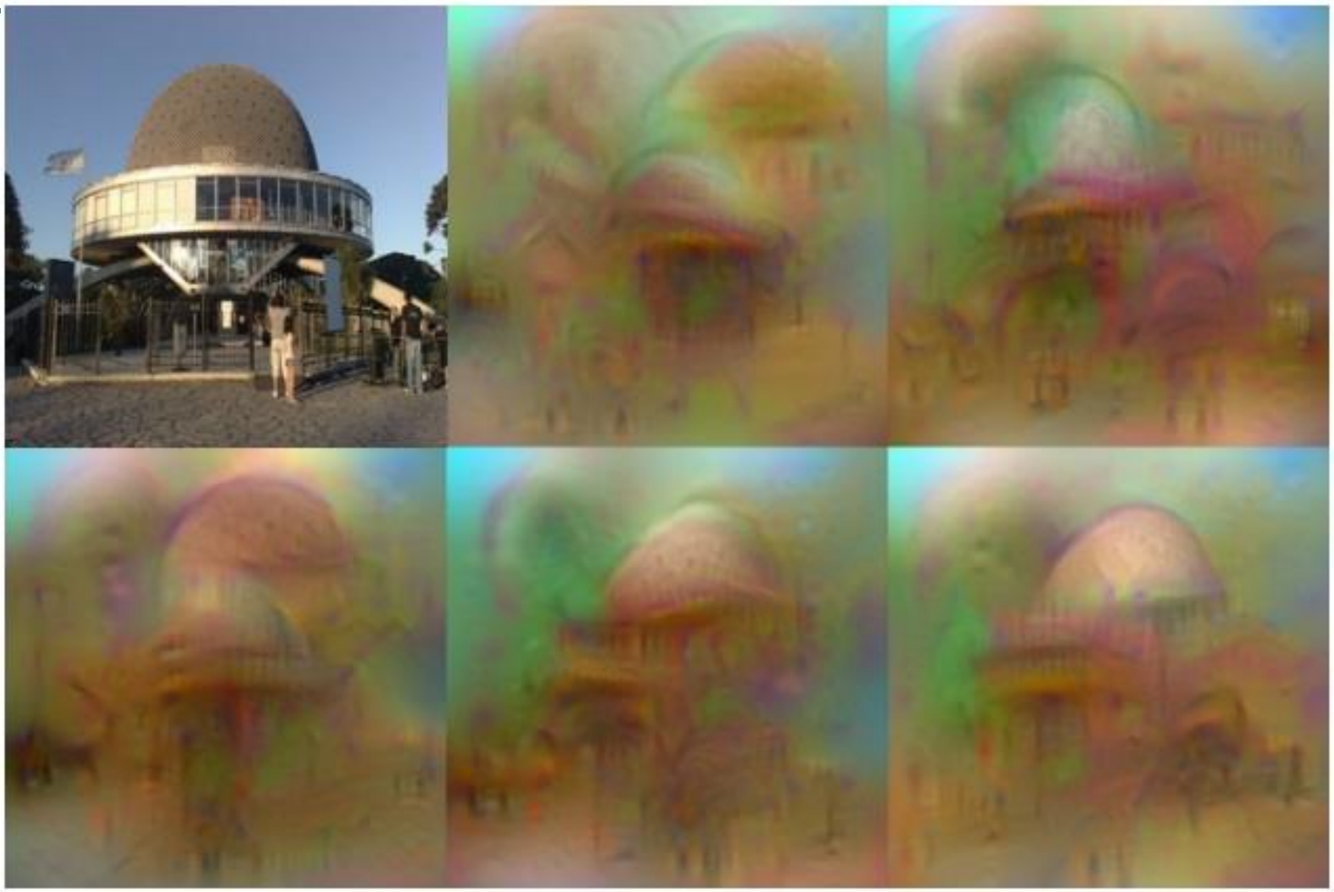
Layer 4 and 5



Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Invert CNN features

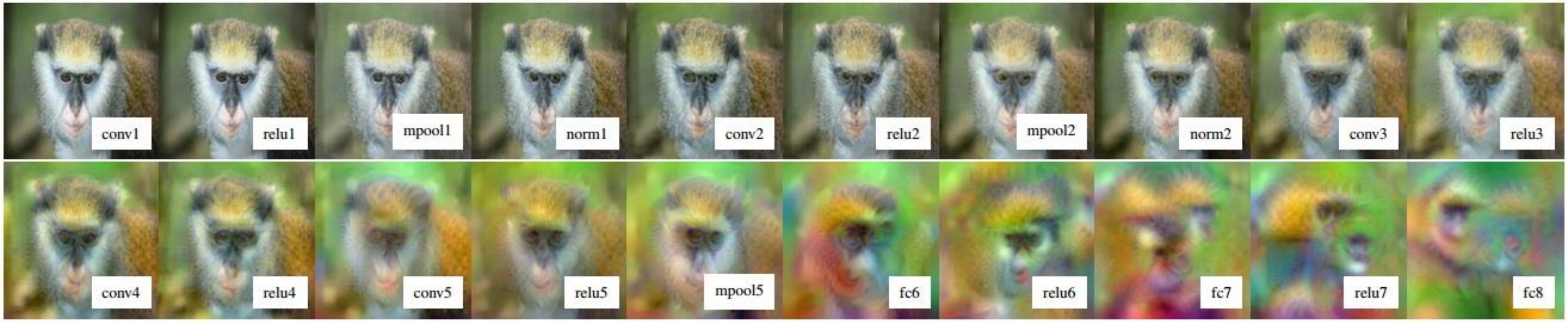
- Reconstruct an image from CNN features



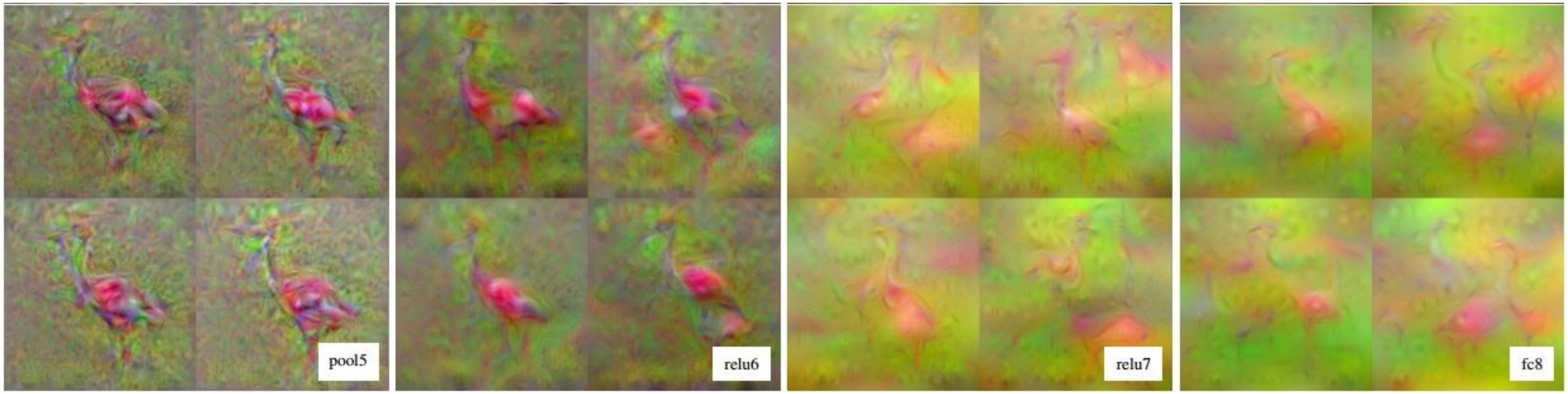
Understanding deep image representations by inverting them
[Mahendran and Vedaldi CVPR 2015]

CNN Reconstruction

Reconstruction from different layers

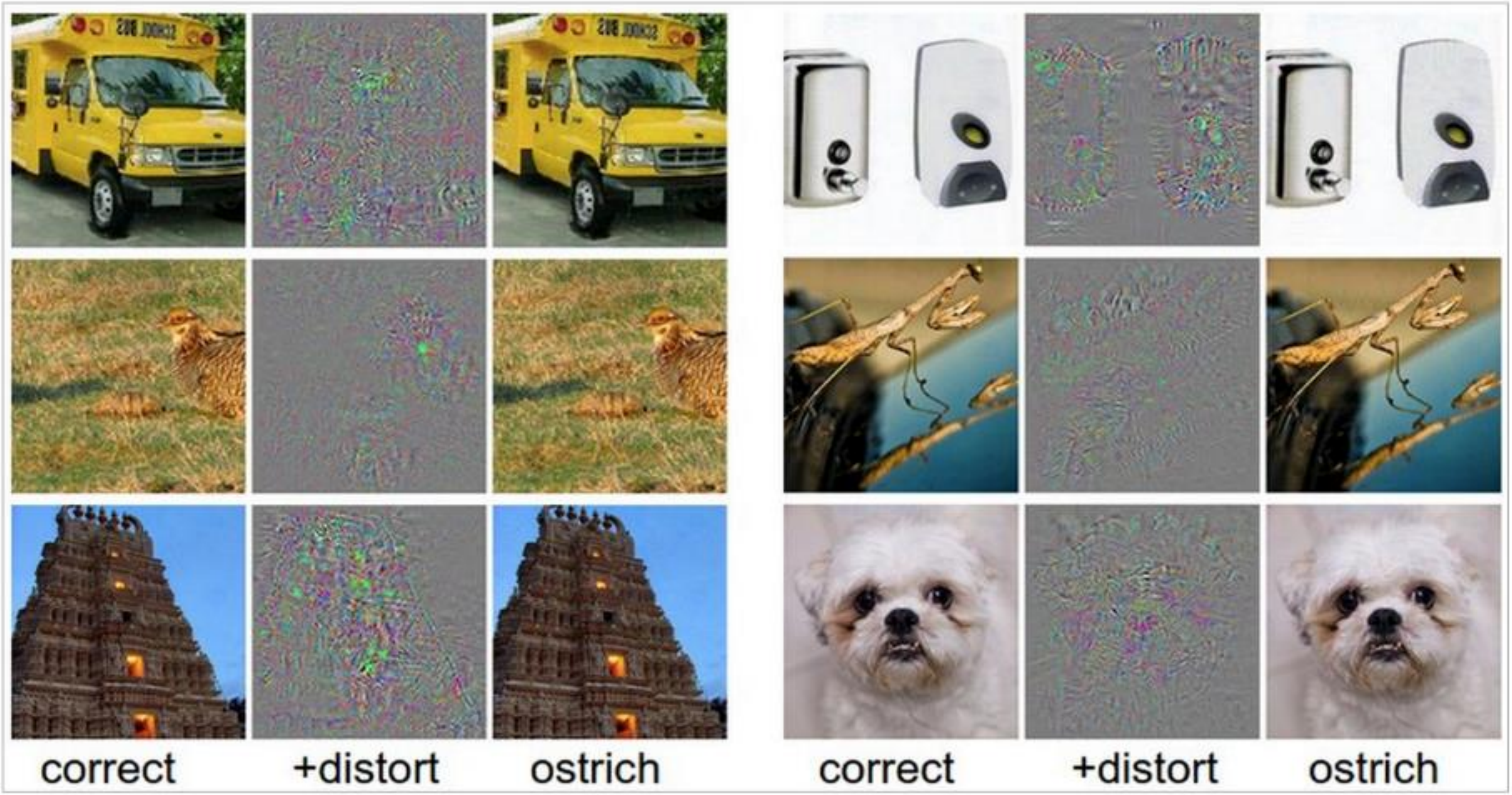


Multiple reconstructions



Understanding deep image representations by inverting them
[Mahendran and Vedaldi CVPR 2015]

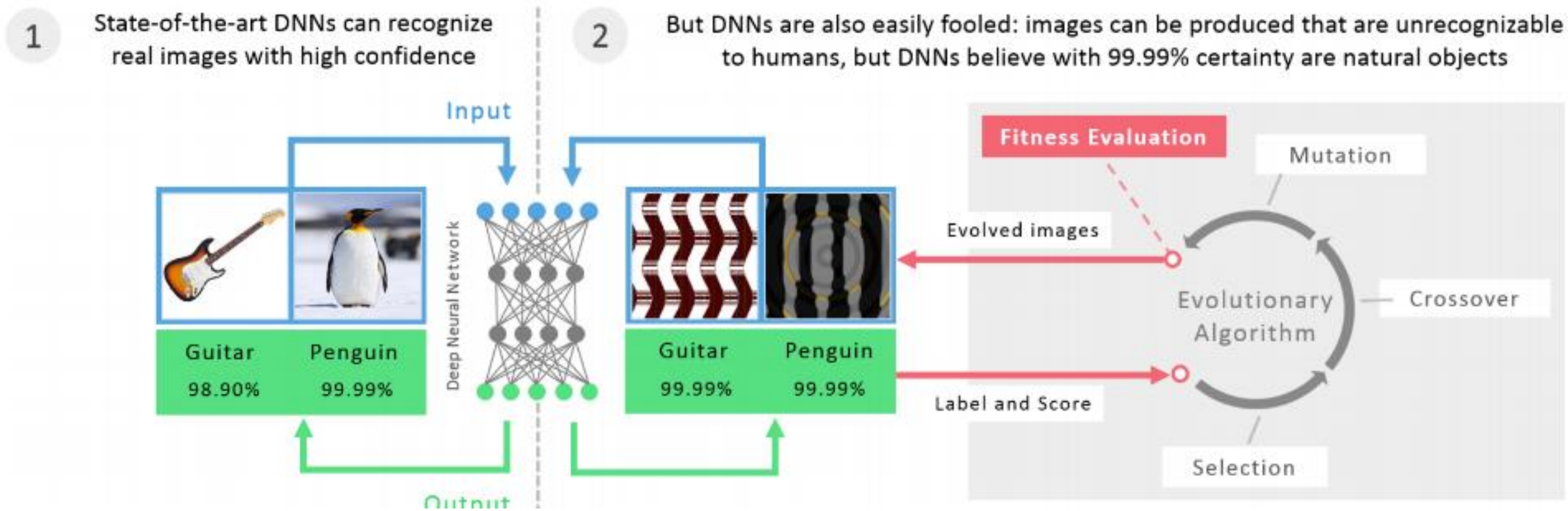
Breaking CNNs



Take a correctly classified image (left image in both columns), and add a tiny distortion (middle) to fool the ConvNet with the resulting image (right).

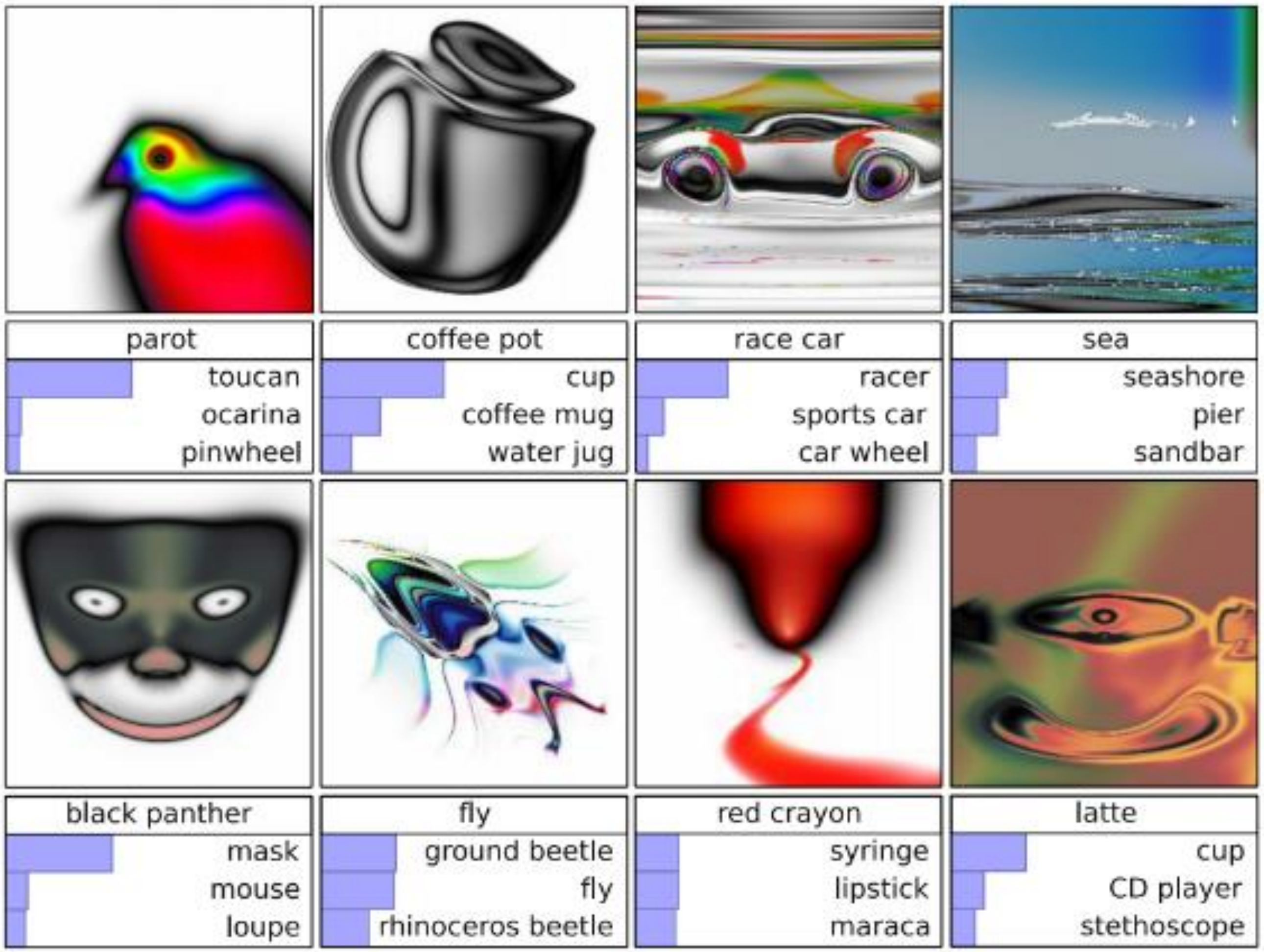
Intriguing properties of neural networks [Szegedy ICLR 2014]

Breaking CNNs



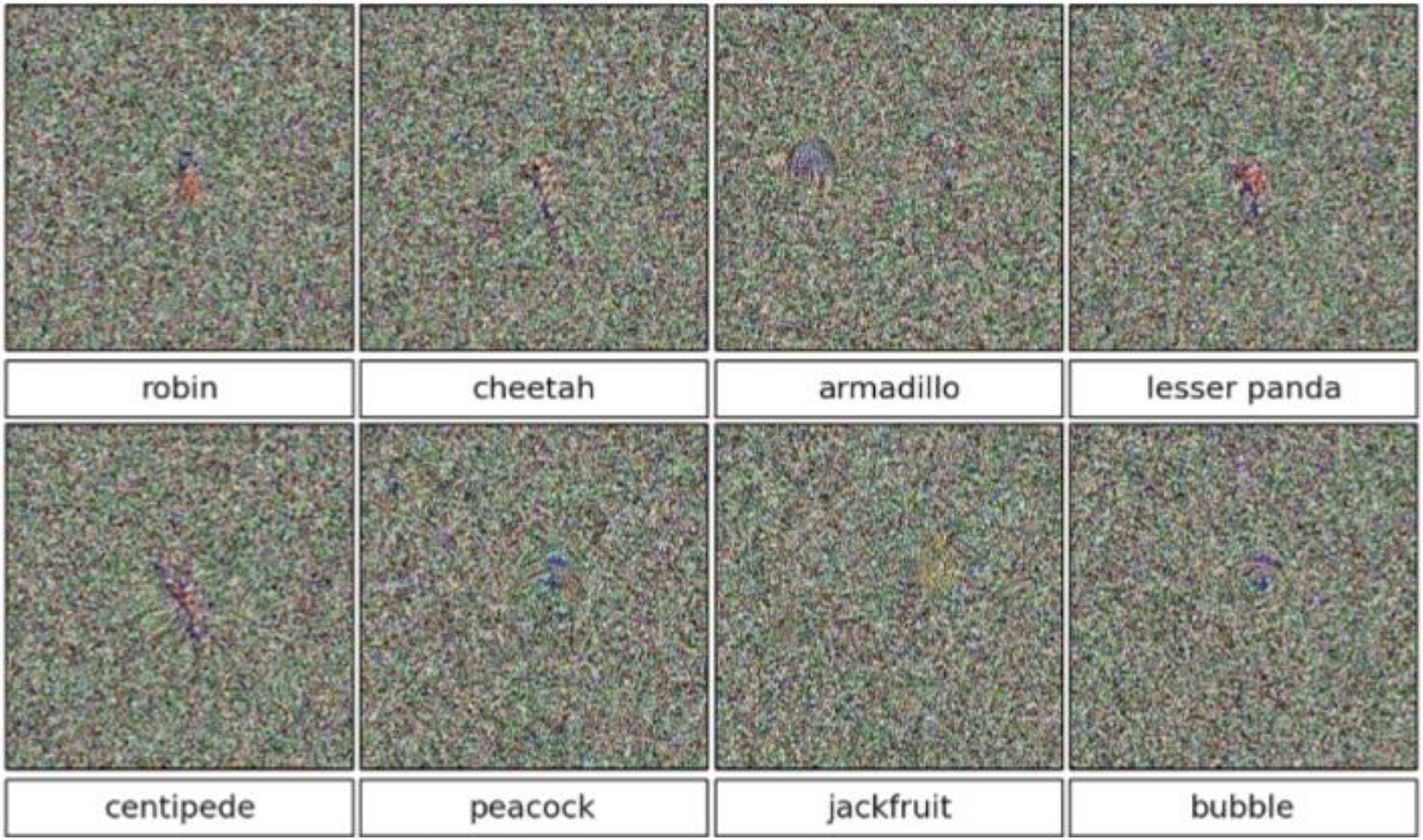
Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

Images that both CNN and Human can recognize



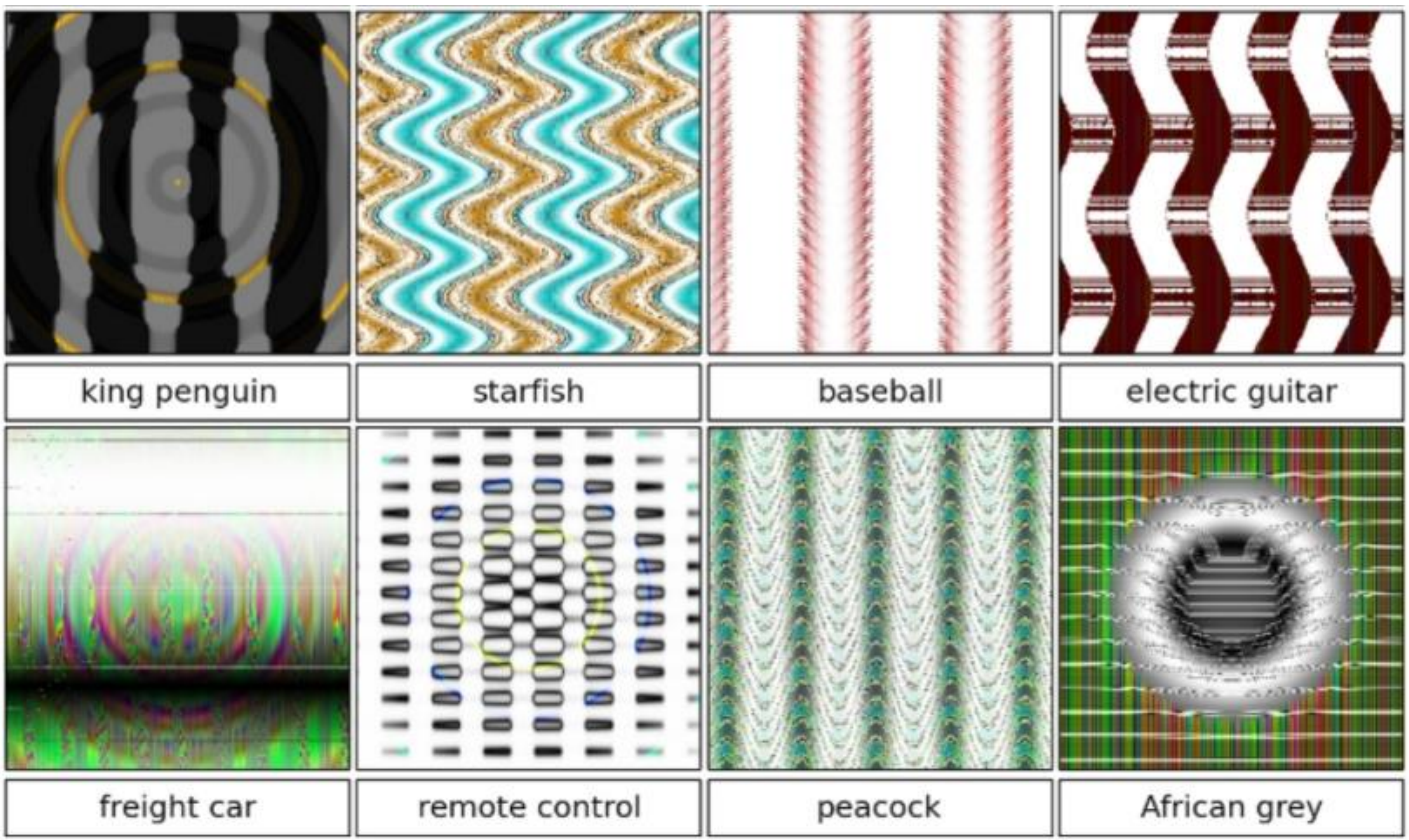
Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

Direct Encoding



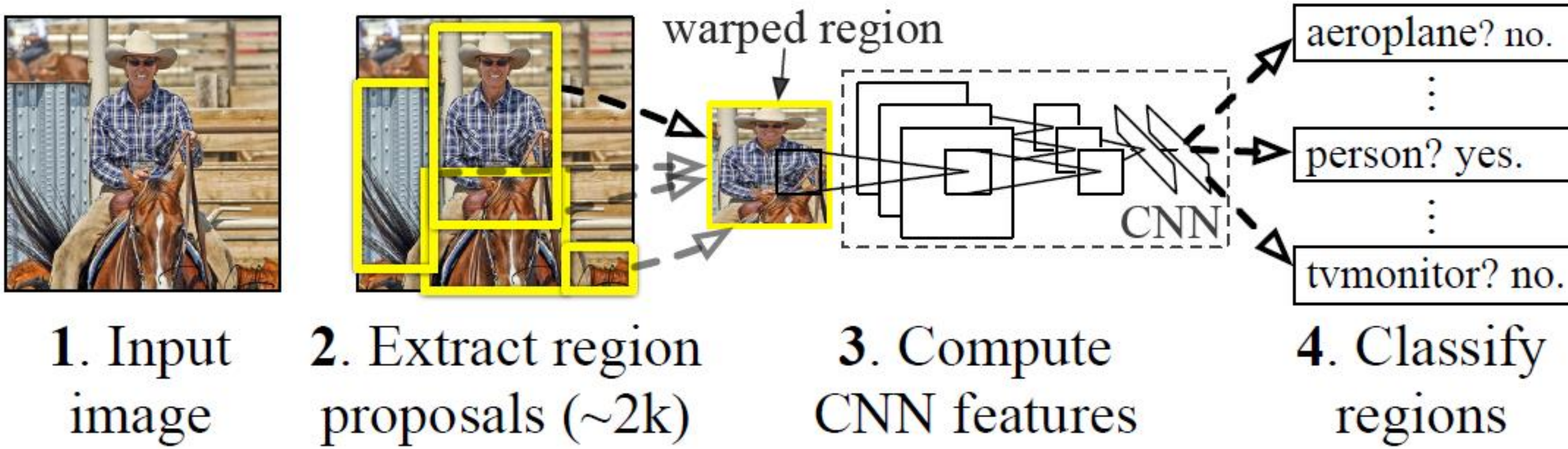
Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

Indirect Encoding



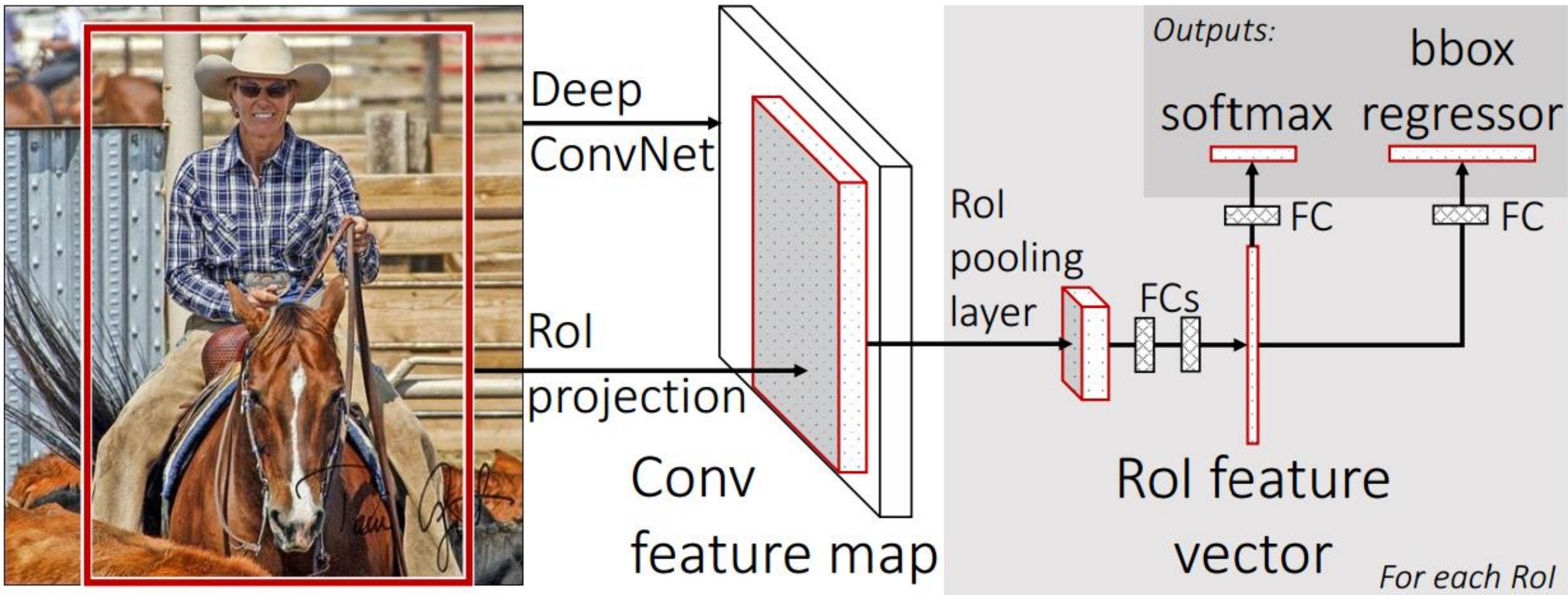
Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

R-CNN (Girshick et al. CVPR 2014)



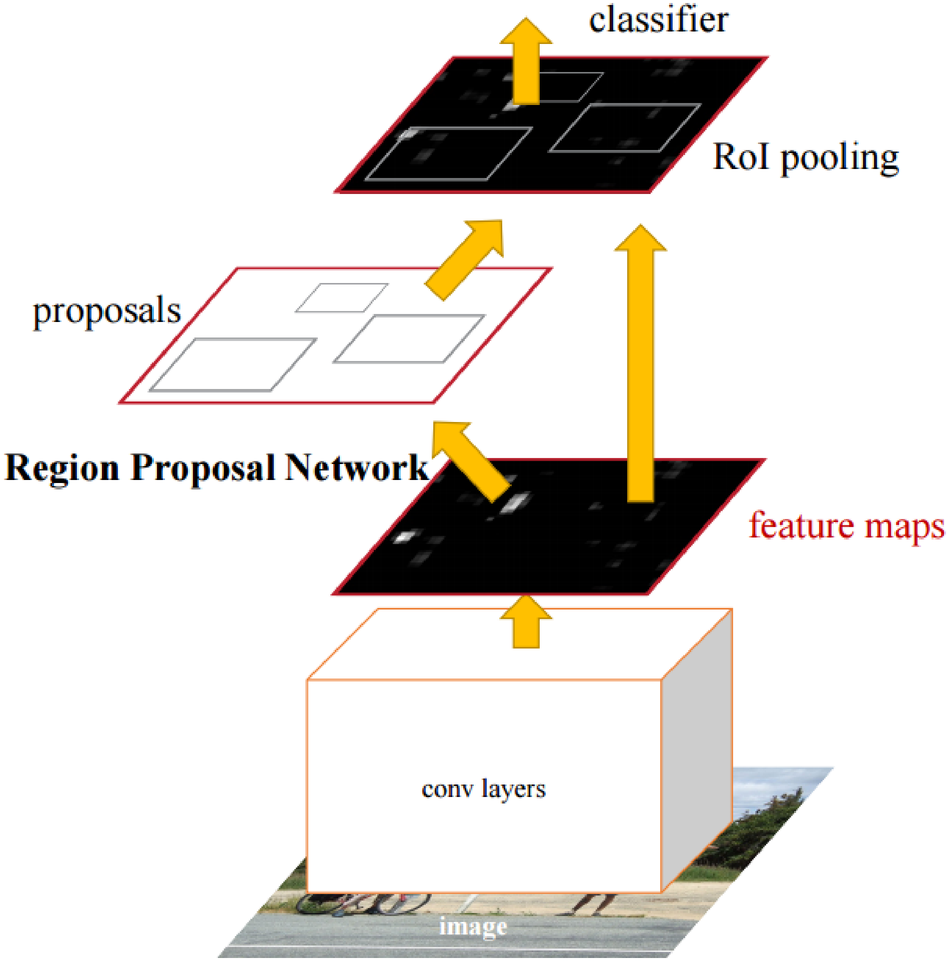
- Replace sliding windows with "selective search" region proposals (Uijilings et al. IJCV 2013)
- Extract rectangles around regions and resize to 227x227
- Extract features with fine-tuned CNN (that was initialized with network trained on ImageNet before training)
- Classify last layer of network features with SVM

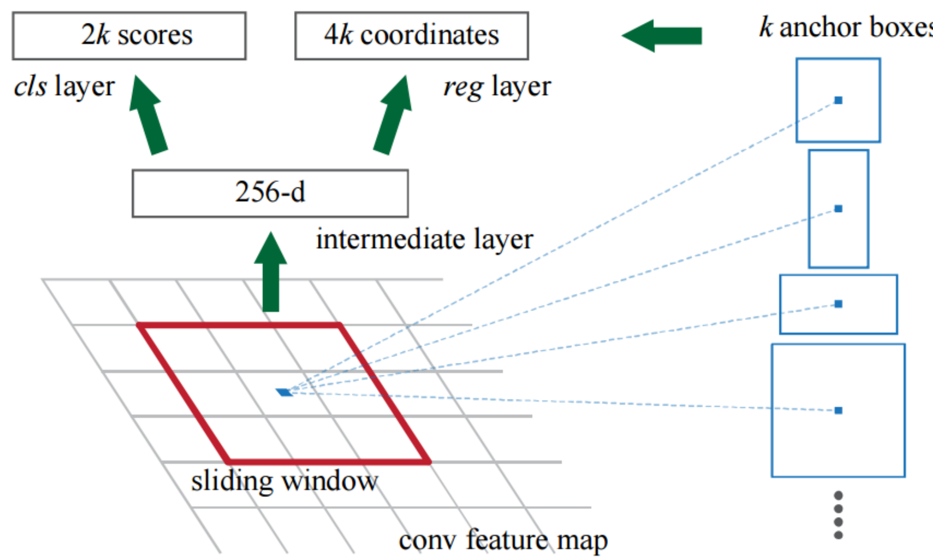
http://arxiv.org/pdf/1311.2524.pdf



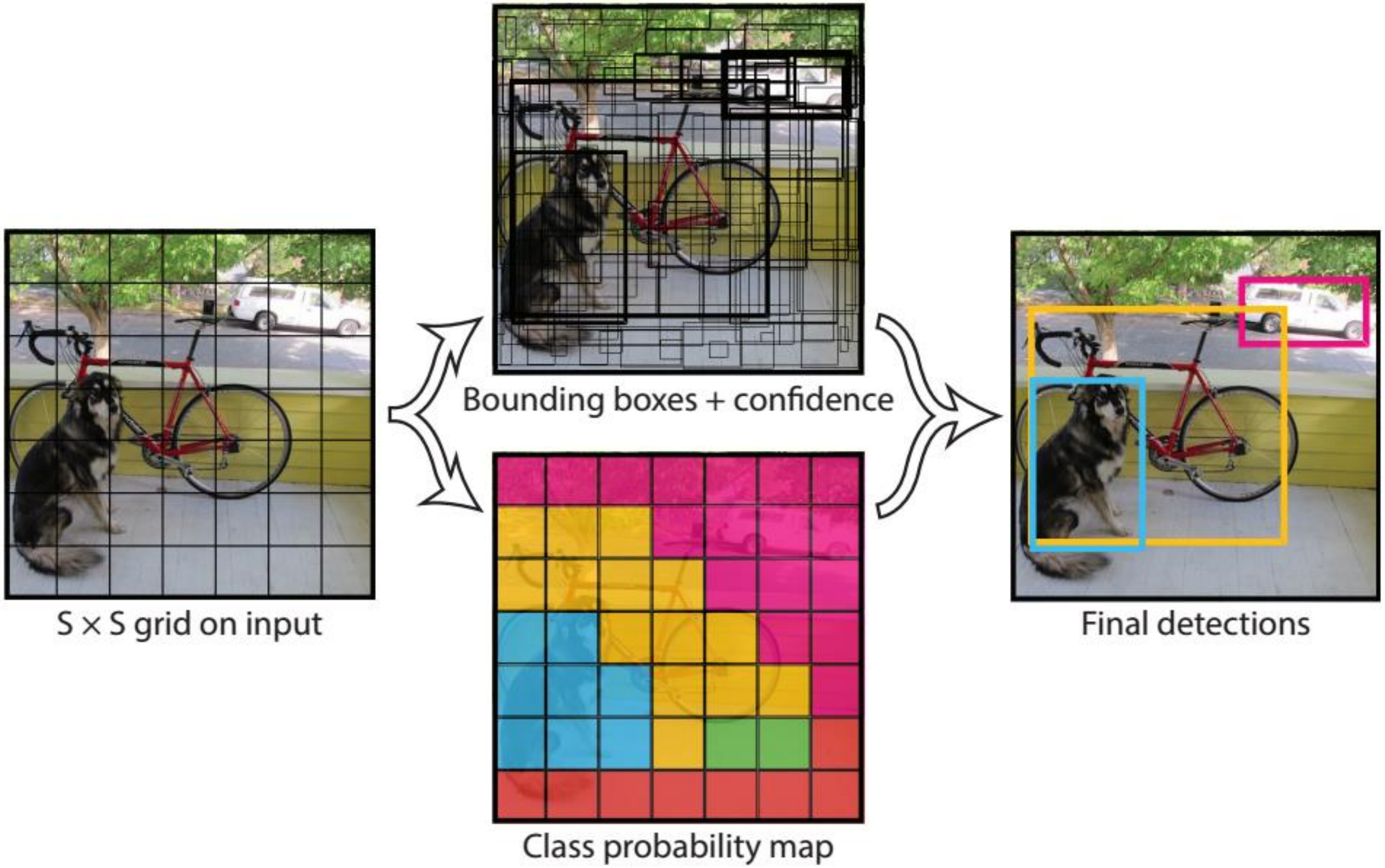
method	train set	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv	mAP
SPPnet BB [11]†	07 $\setminus$ diff	73.9	72.3	62.5	51.5	44.4	74.4	73.0	74.4	42.3	73.6	57.7	70.3	74.6	74.3	54.2	34.0	56.4	56.4	67.9	73.5	63.1
R-CNN BB [10]	07	73.4	77.0	63.4	45.4	**44.6**	75.1	78.1	79.8	40.5	73.7	62.2	79.4	78.1	73.1	64.2	**35.6**	66.8	67.2	70.4	**71.1**	66.0
FRCN [ours]	07	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8	66.9
FRCN [ours]	07 $\setminus$ diff	74.6	**79.0**	68.6	57.0	39.3	79.5	**78.6**	81.9	**48.0**	74.0	67.4	80.5	80.7	74.1	69.6	31.8	67.1	68.4	75.3	65.5	68.1
FRCN [ours]	07+12	**77.0**	78.1	**69.3**	**59.4**	38.3	**81.6**	**78.6**	**86.7**	42.8	**78.8**	**68.9**	**84.7**	**82.0**	**76.6**	**69.9**	31.8	**70.1**	**74.8**	**80.4**	70.4	**70.0**

Fast RCNN, ICCV 2015

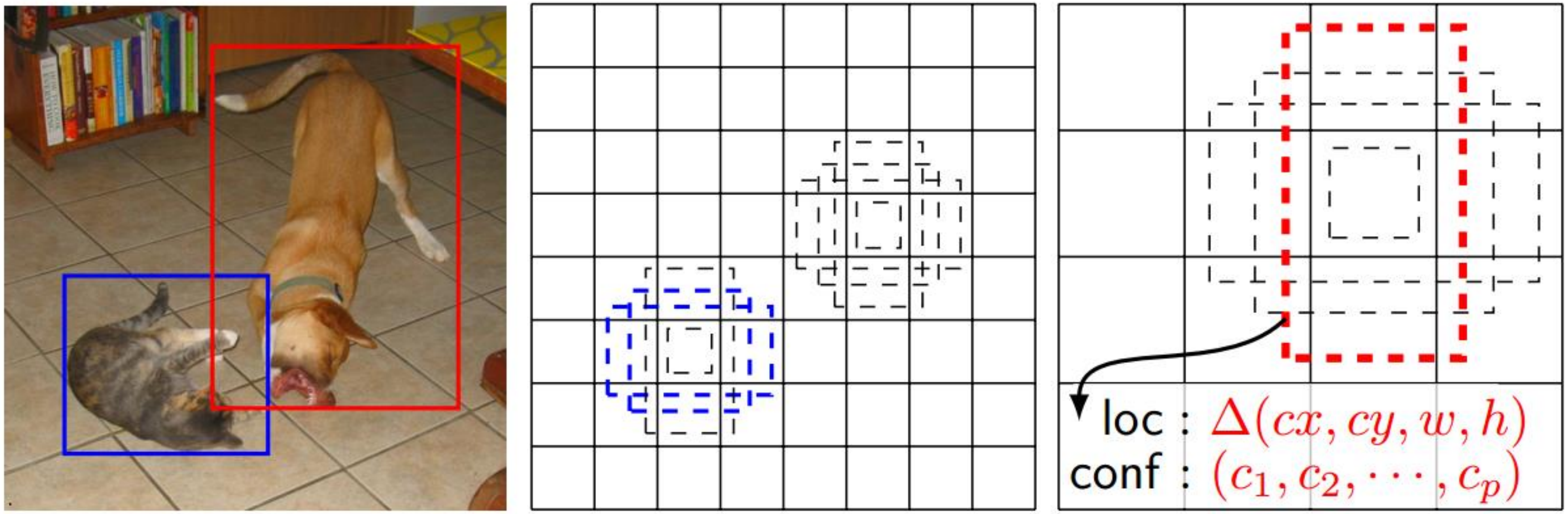




[Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015]()



YOLO: Real-Time Object Detection, CVPR 2016

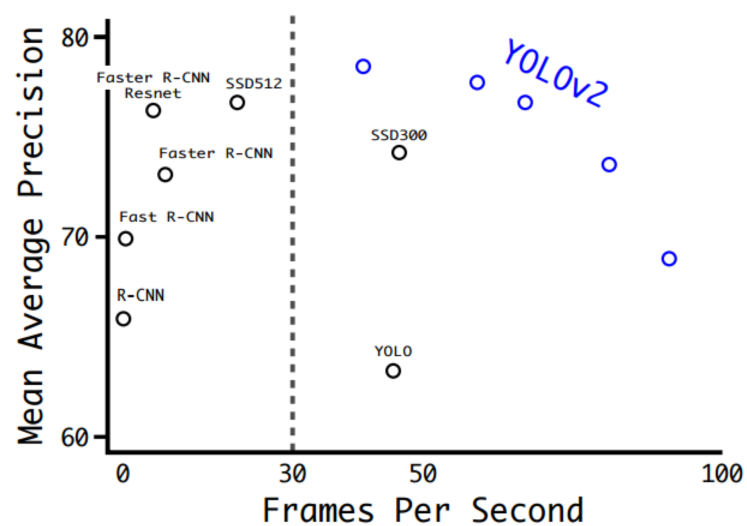


(a) Image with GT boxes (b) 8 × 8 feature map (c) 4 × 4 feature map

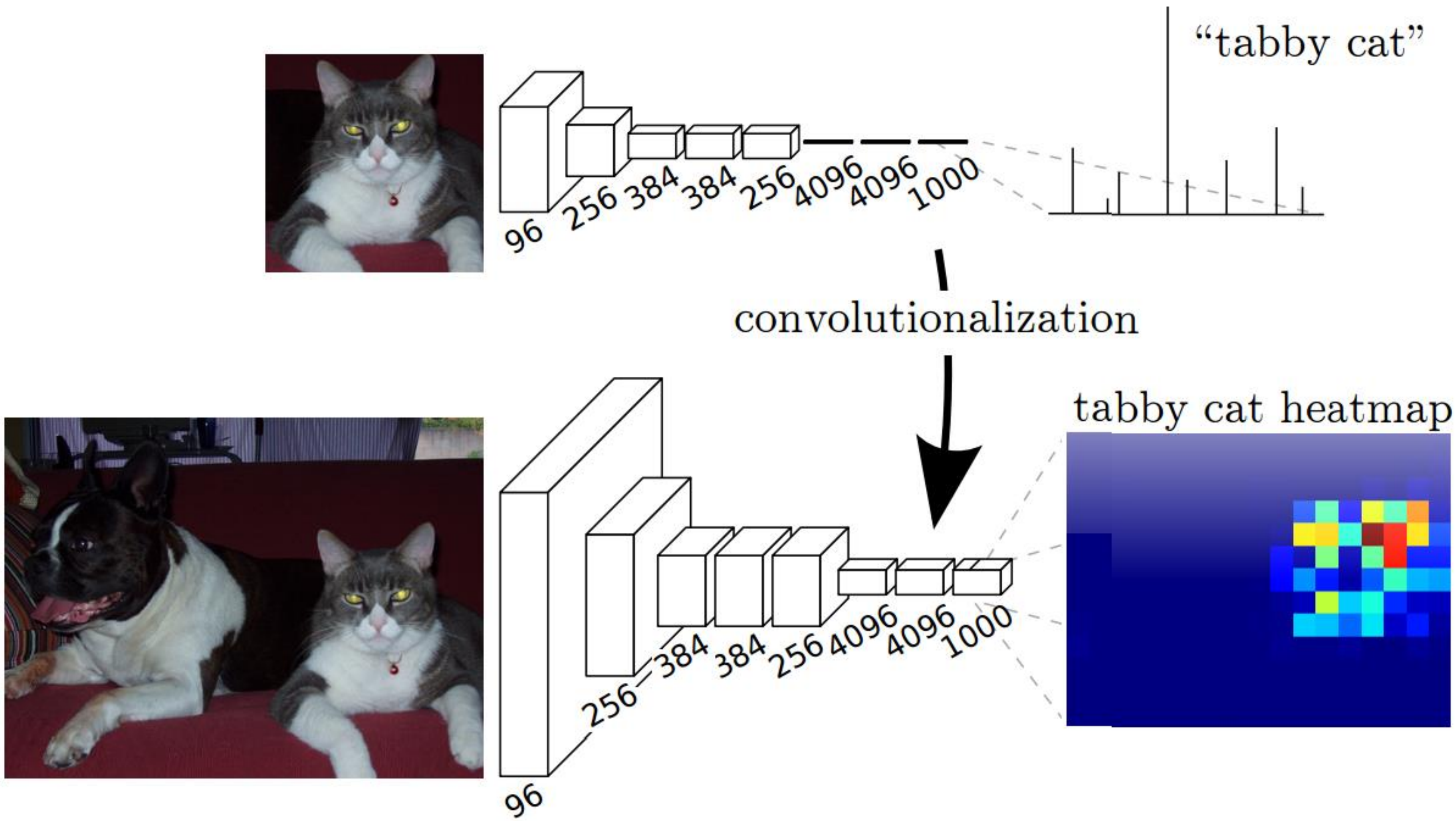
Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	∼ 6000	∼ 1000 × 600
Fast YOLO	52.7	155	1	98	448 × 448
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8732	300 × 300
SSD512	76.8	19	1	24564	512 × 512
SSD300	74.3	59	8	8732	300 × 300
SSD512	76.8	22	8	24564	512 × 512

[SSD: Single Shot MultiBox Detector, ECCV 2016]()

	YOLO								YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

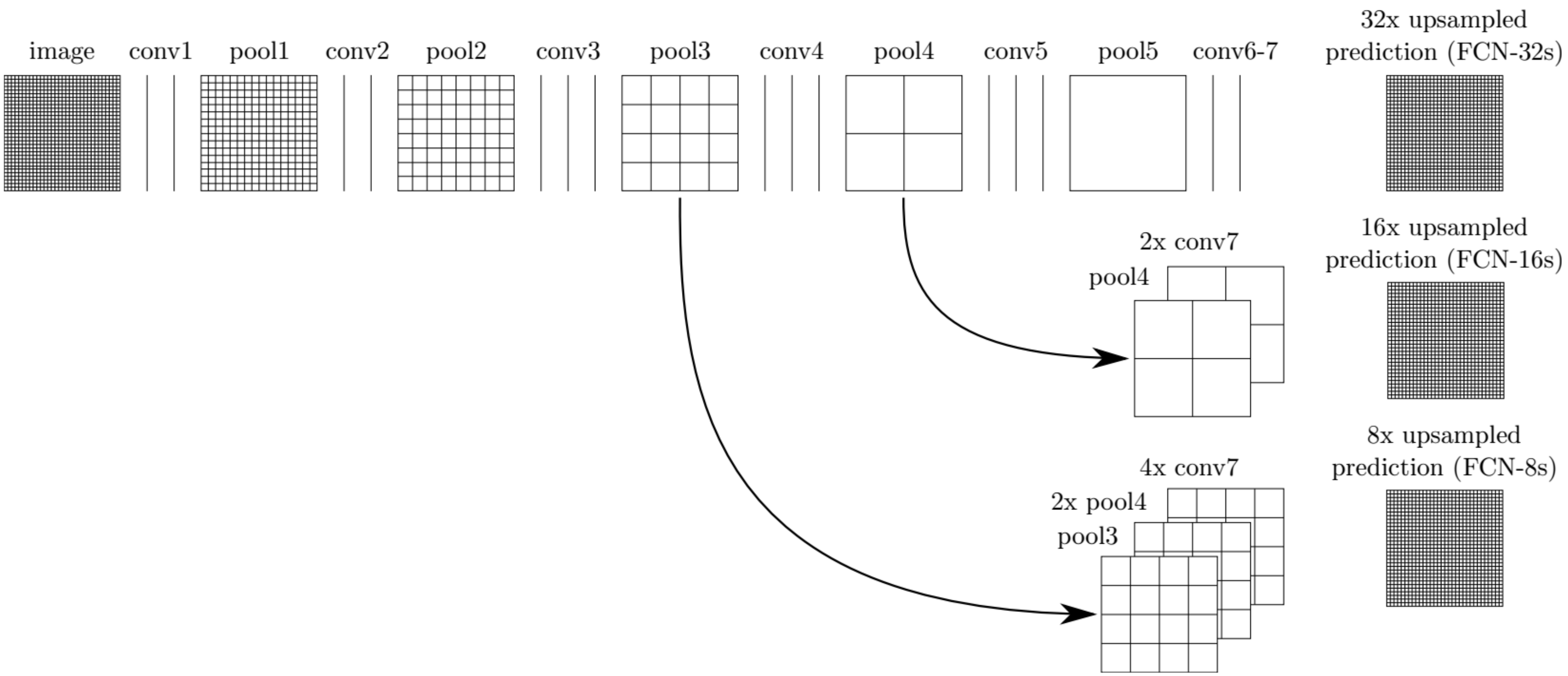


YOLO9000: Better, Faster, Stronger, arXiv 2016

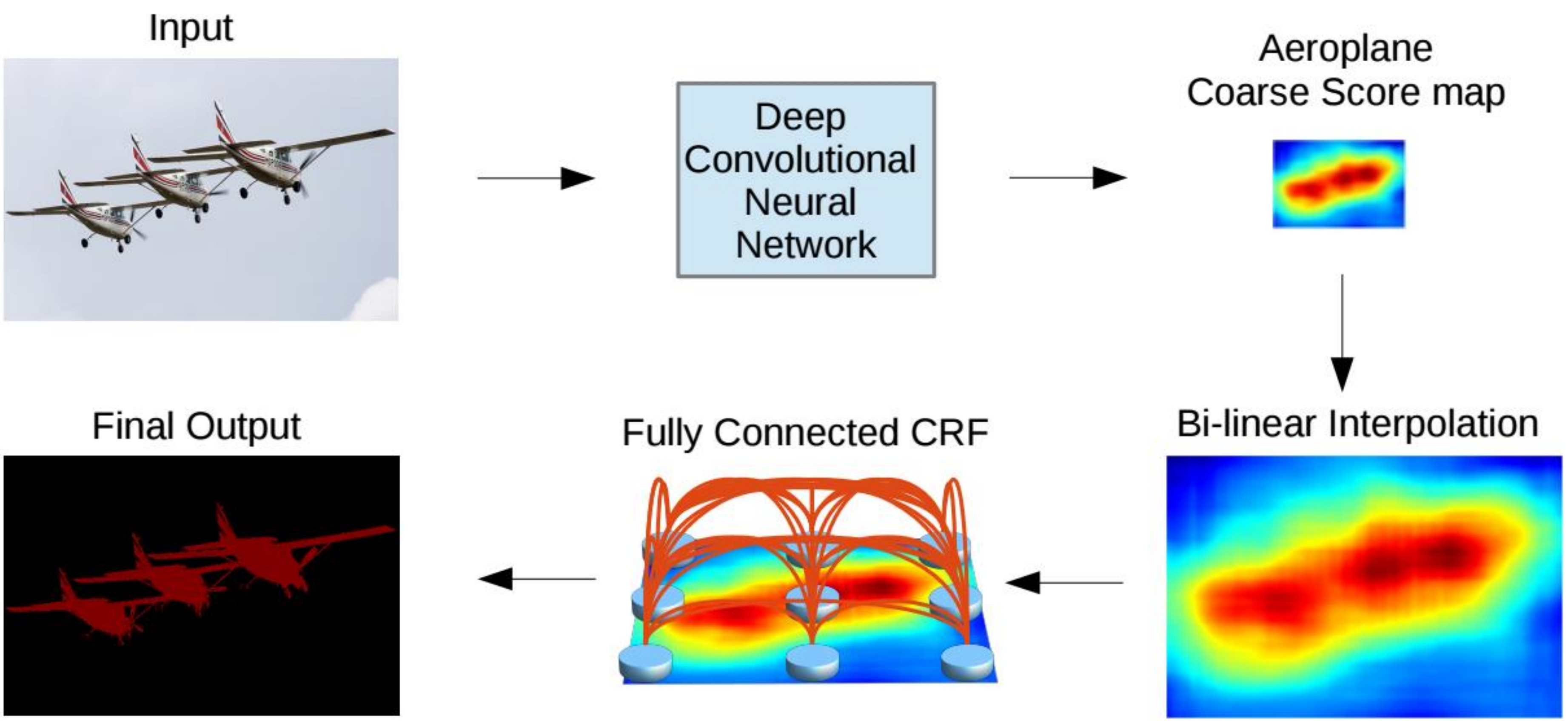


[Fully Convolutional Networks for Semantic Segmentation, CVPR 2015]

Combining what and where

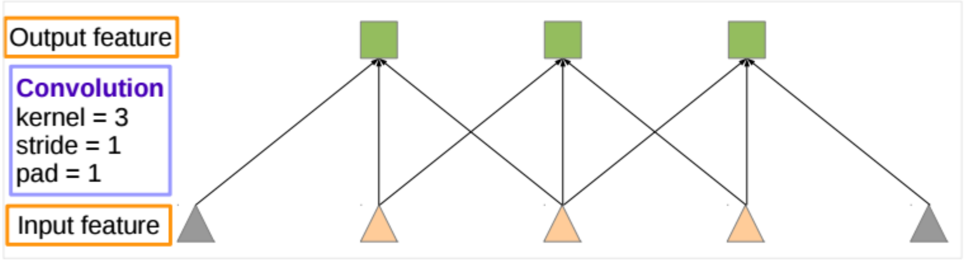


[Fully Convolutional Networks for Semantic Segmentation, CVPR 2015]()

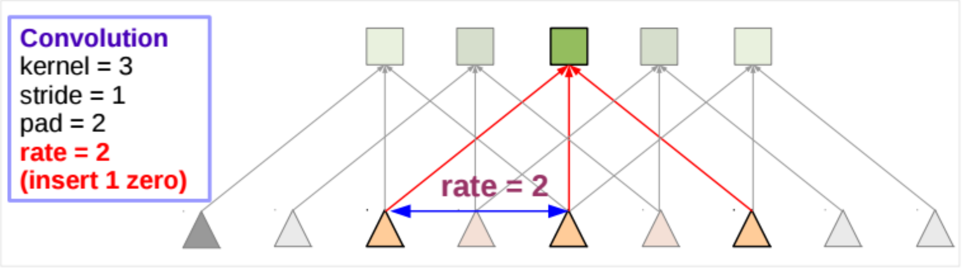


[DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, 2016]

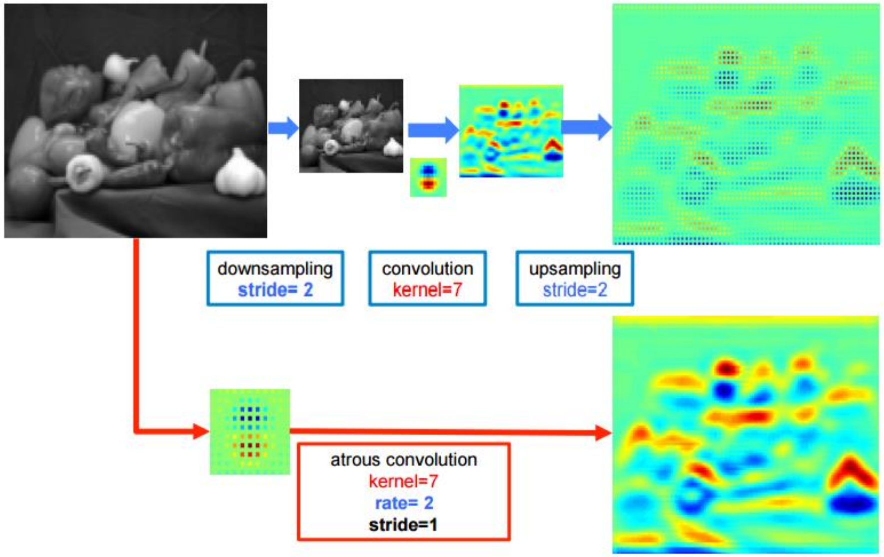
Atrous convolution



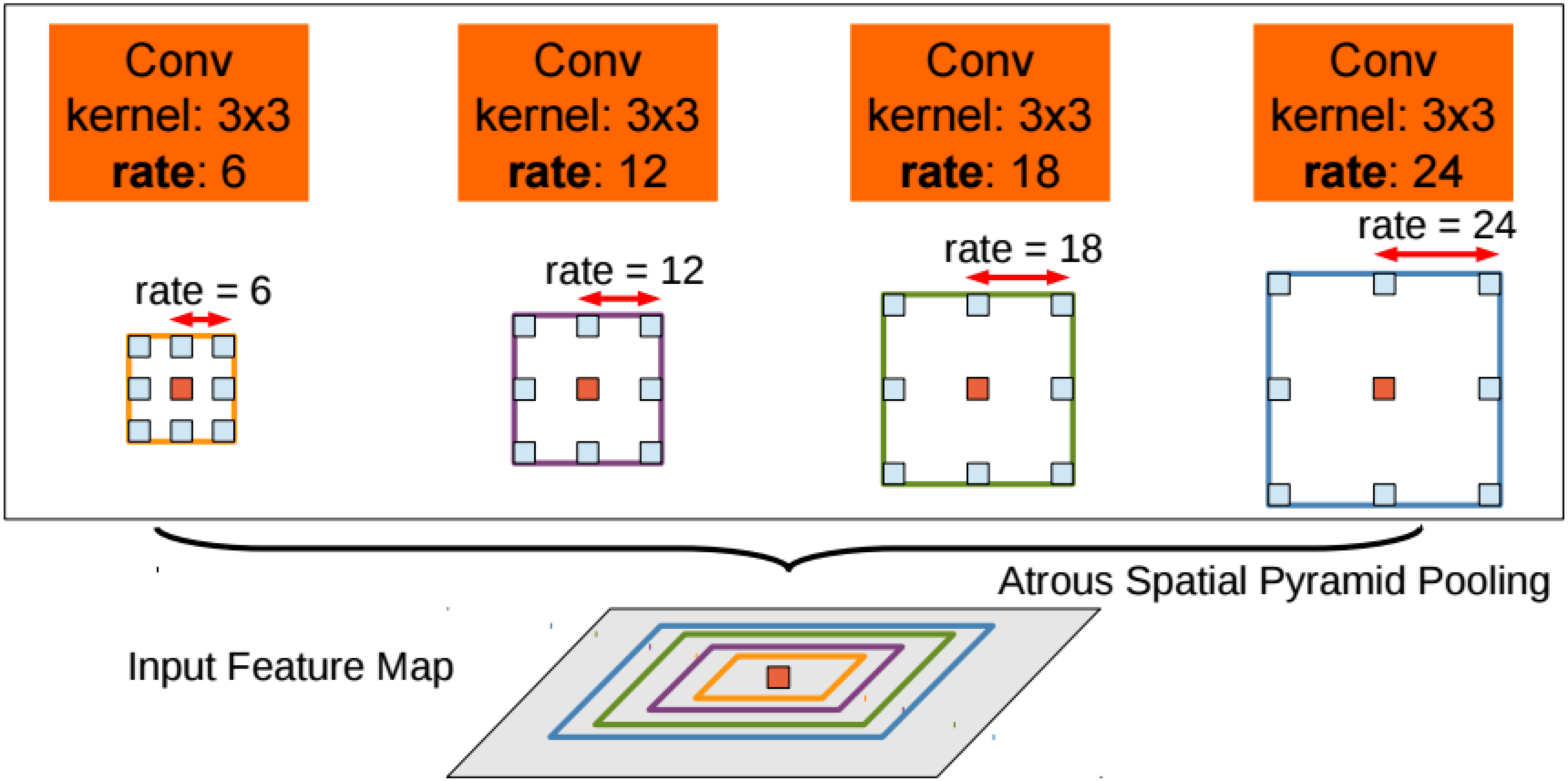
(a) Sparse feature extraction



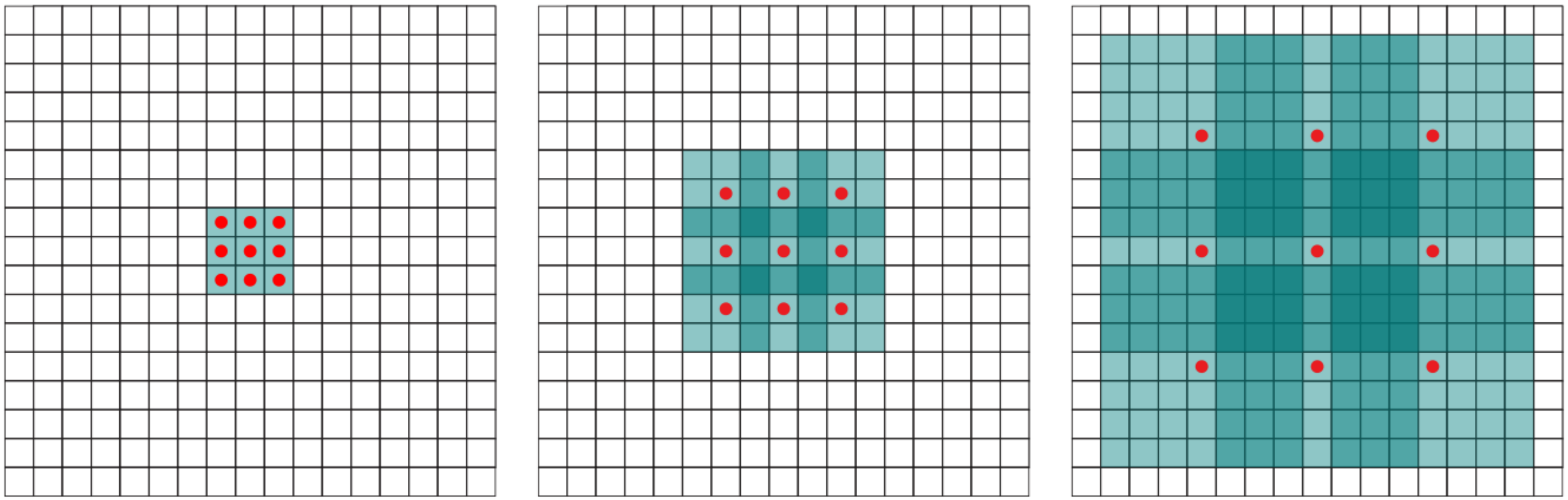
(b) Dense feature extraction



Atrous spatial pyramid pooling



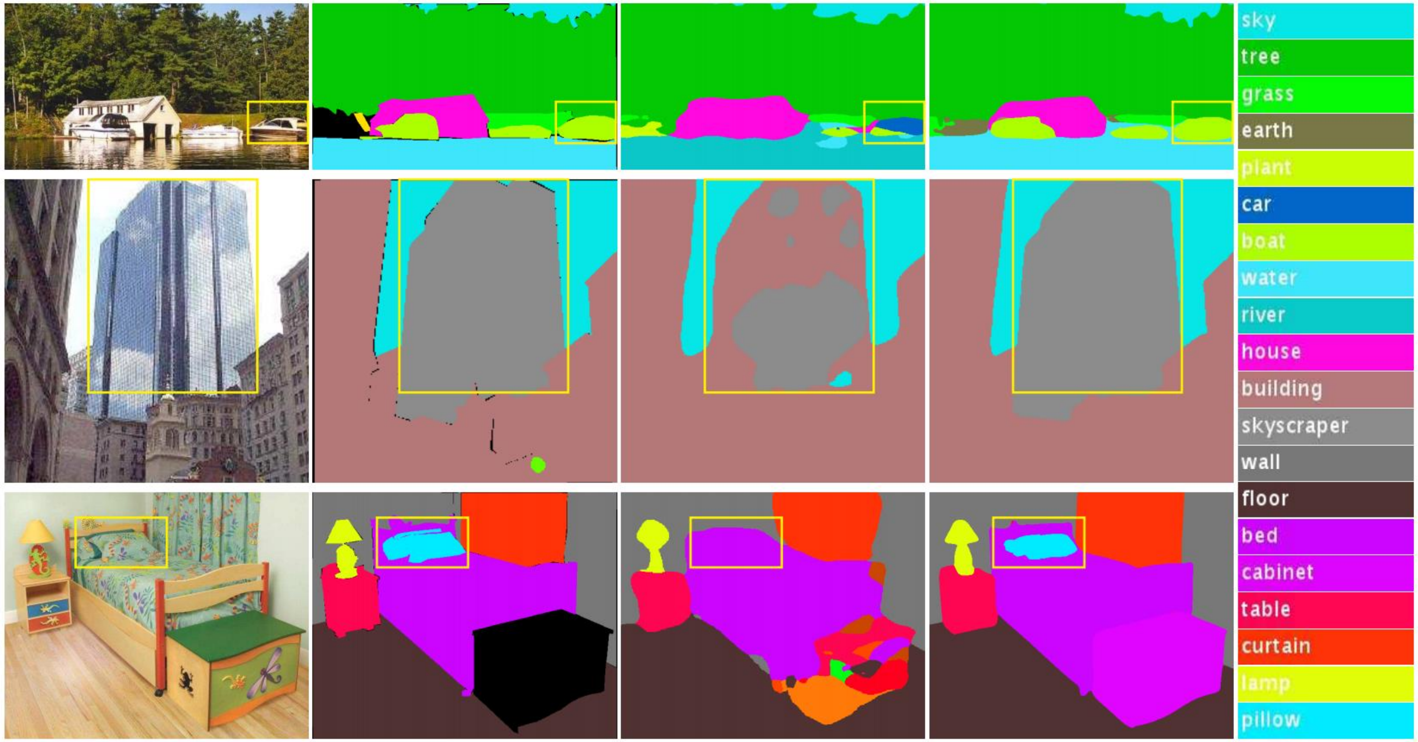
Dilated convolution



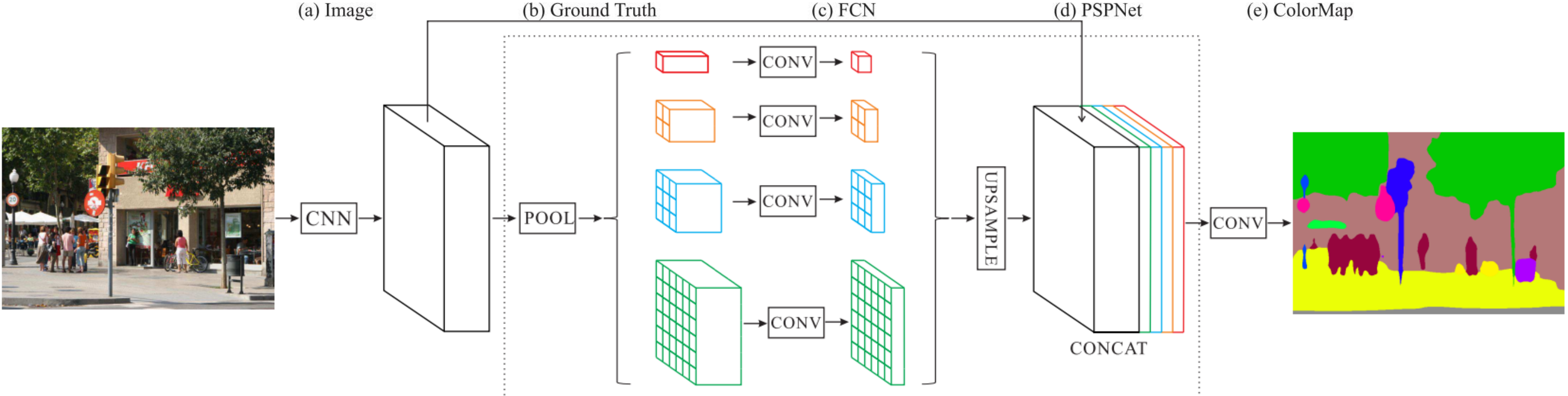
(a) (b) (c)

Figure 1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) F_1 is produced from F_0 by a 1-dilated convolution; each element in F_1 has a receptive field of 3×3. (b) F_2 is produced from F_1 by a 2-dilated convolution; each element in F_2 has a receptive field of 7×7. (c) F_3 is produced from F_2 by a 4-dilated convolution; each element in F_3 has a receptive field of 15×15. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

[Multi-scale Context Aggregation by Dilated Convolutions, ICLR 2016]



(a) Image (b) Ground Truth (c) FCN (d) PSPNet (e) ColorMap

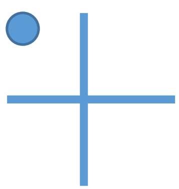
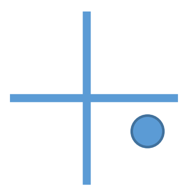


(a) Input Image (b) Feature Map (c) Pyramid Pooling Module (d) Final Prediction

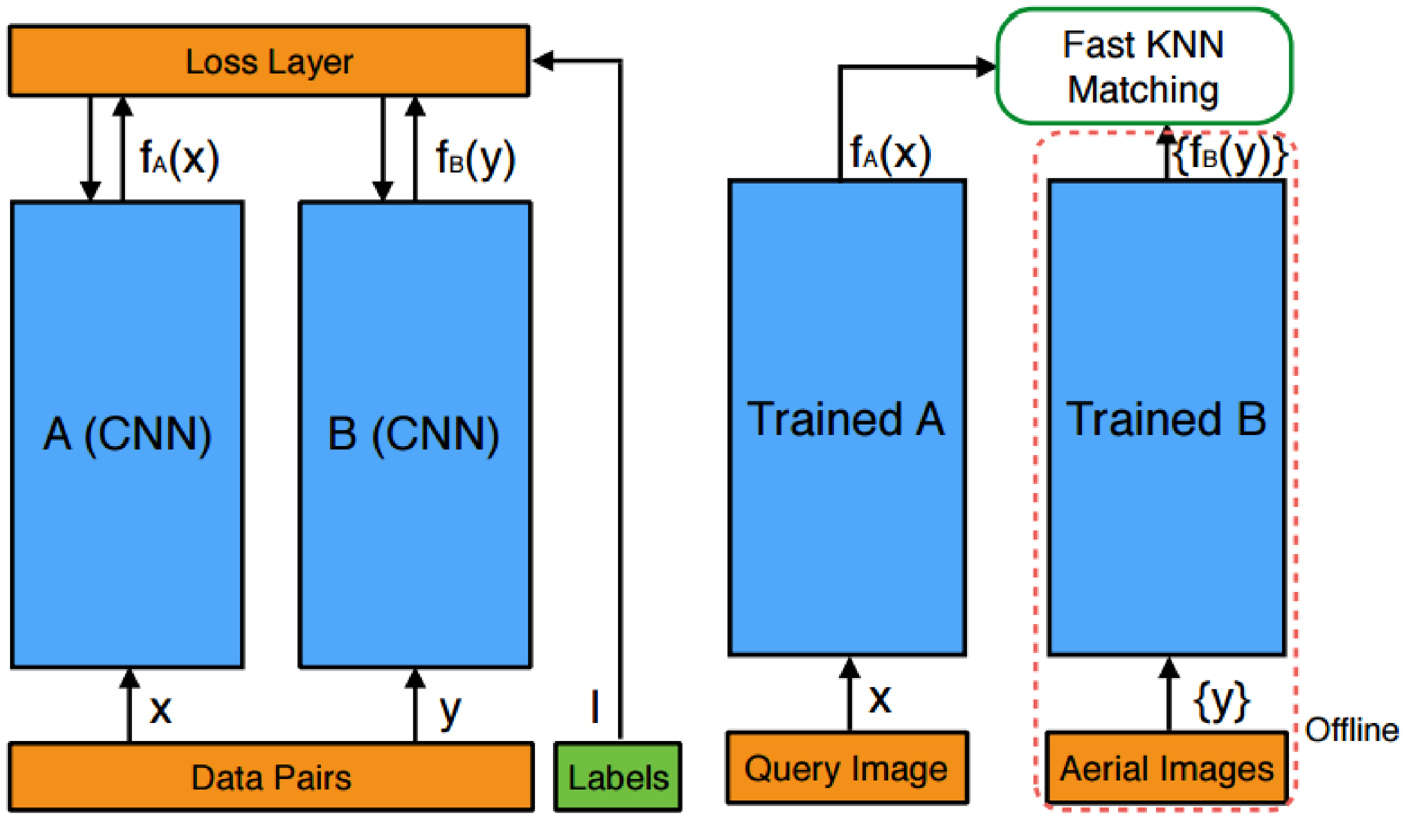
[Pyramid Scene Parsing Network, 2016]

Pyramid Scene Parsing Network, 2016

Siamese/Triplet networks for distance metric learning

- Distance metric learning
  - Learn feature embedding so that the distances capture the semantic similarity
- D(‘Tech’, ‘Taco’) = 3
- D( , ) = 25
- D(, ) = 50
- D( , ) = ?

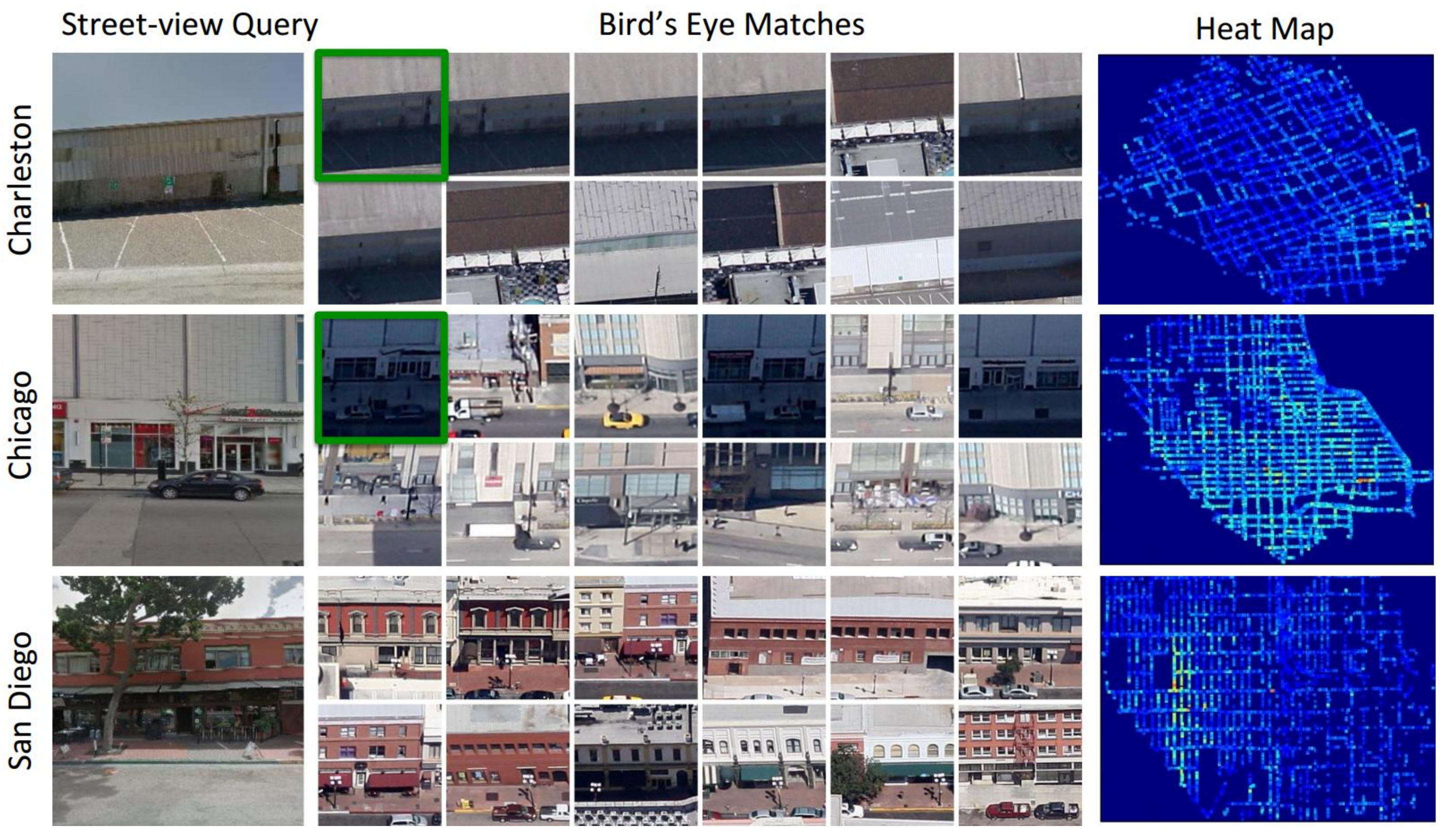
Siamese Network



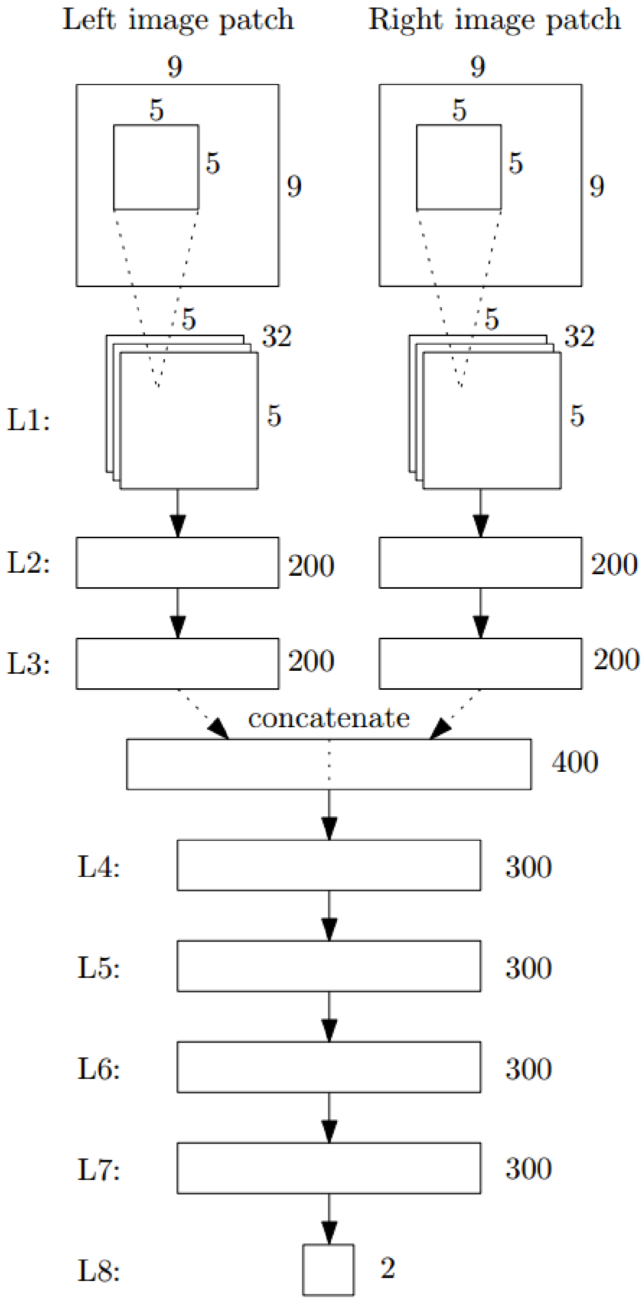
(a) Training (b) Testing

$$\mathcal{L}(x,y,l) = \frac{1}{2} l D^2 + \frac{1}{2}(1-l)\max\left(0, (m - D^2)\right)$$

Learning Deep Representations for Ground-to-Aerial Geolocalization, CVPR 2015



[Learning Deep Representations for Ground-to-Aerial Geolocalization, CVPR 2015]



$$< \mathcal{P}^L_{9\times9}(\mathbf{p}), \mathcal{P}^R_{9\times9}(\mathbf{q}) >$$

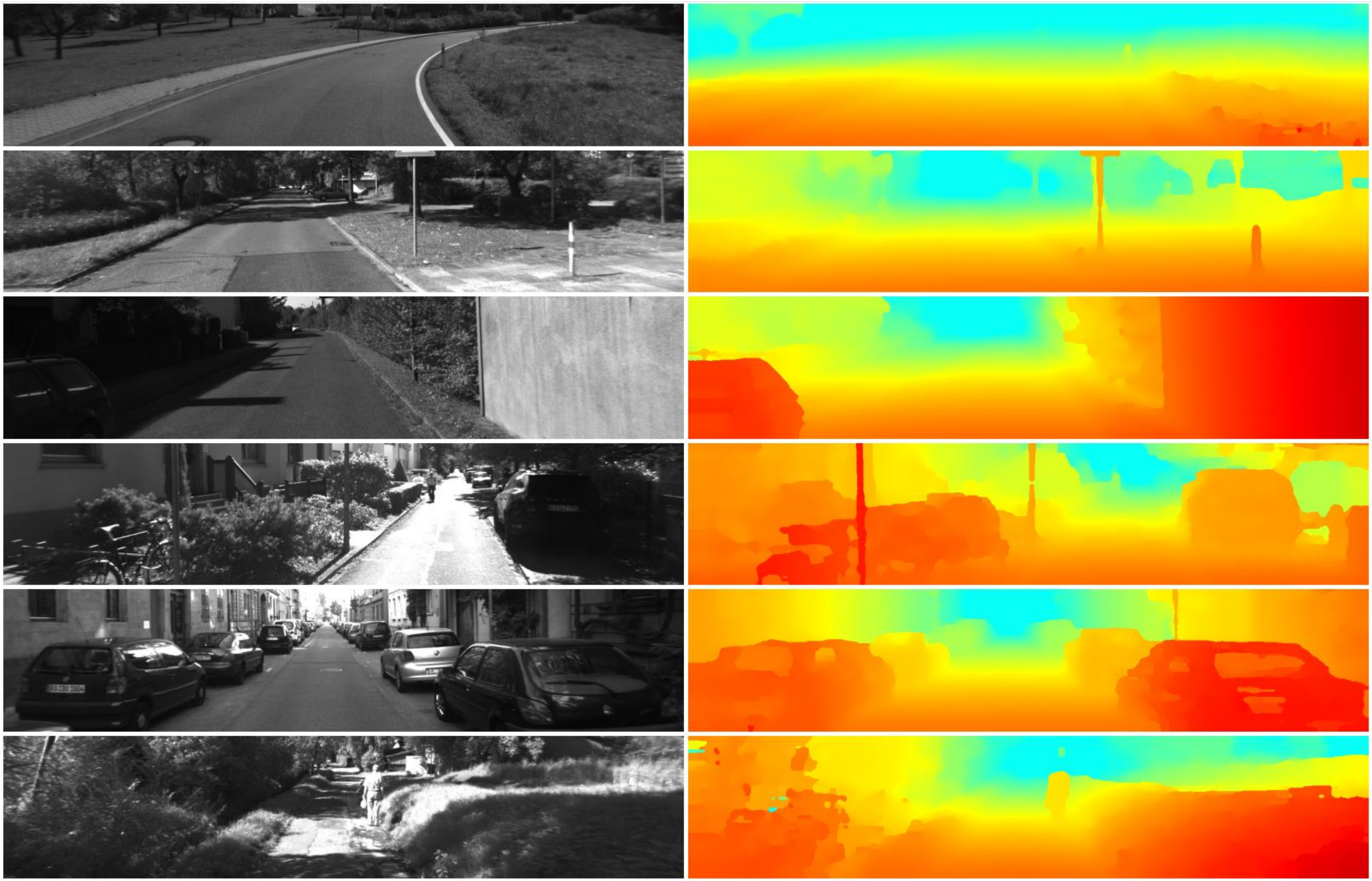
Negative examples

$$\mathbf{q} = (x - d + o_{\text{neg}}, y)$$

Positive examples

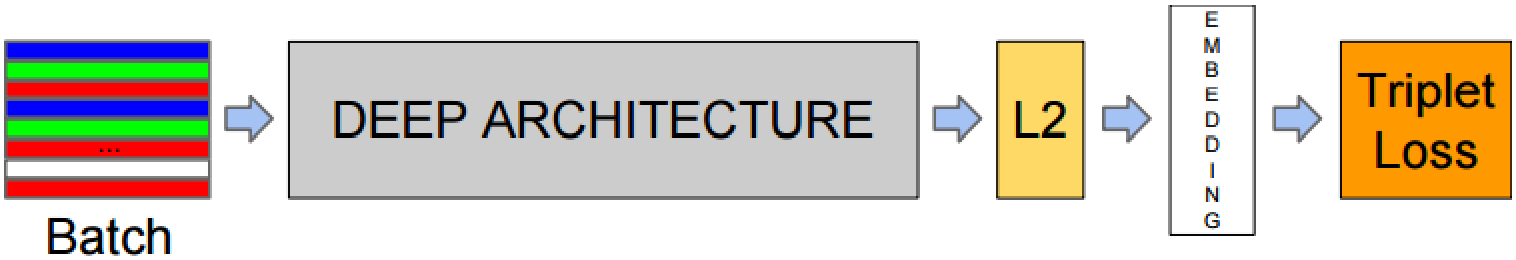
$$\mathbf{q} = (x - d + o_{\text{pos}}, y)$$

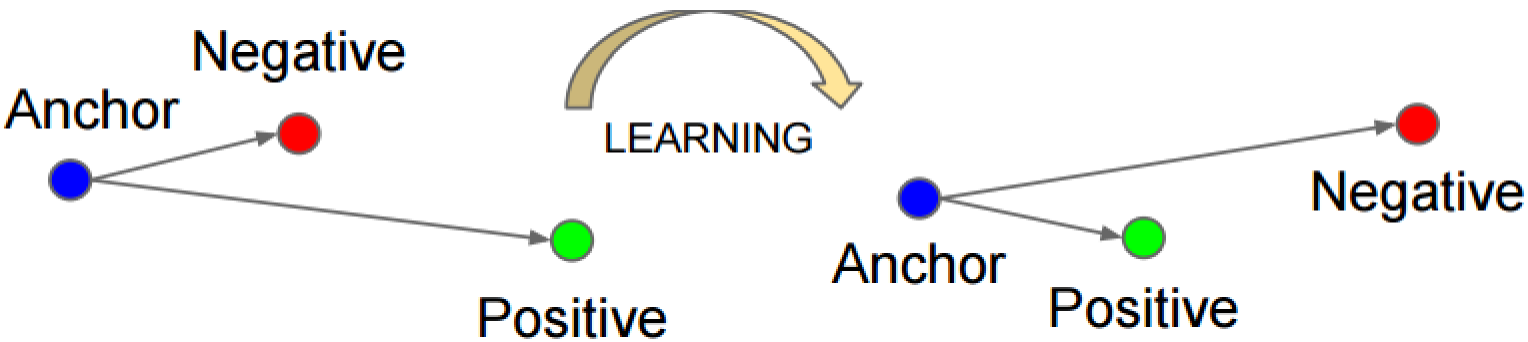
[Computing the Stereo Matching Cost with a Convolutional Neural Network, CVPR 2015]



[Computing the Stereo Matching Cost with a Convolutional Neural Network, CVPR 2015]

Triplet loss

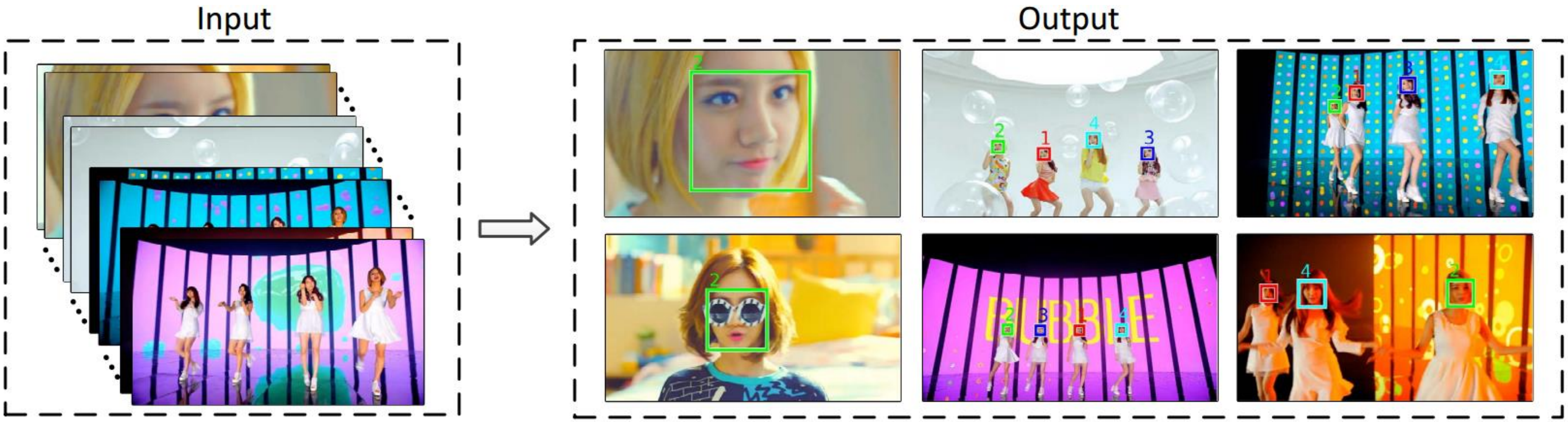




$$\sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

[FaceNet: A Unified Embedding for Face Recognition and Clustering, CVPR 2015]

Problem: Multi-face tracking



[Tracking Persons-of-Interest via Adaptive Discriminative Features, ECCV 2016]

Major challenge: large appearance variations

Sojin

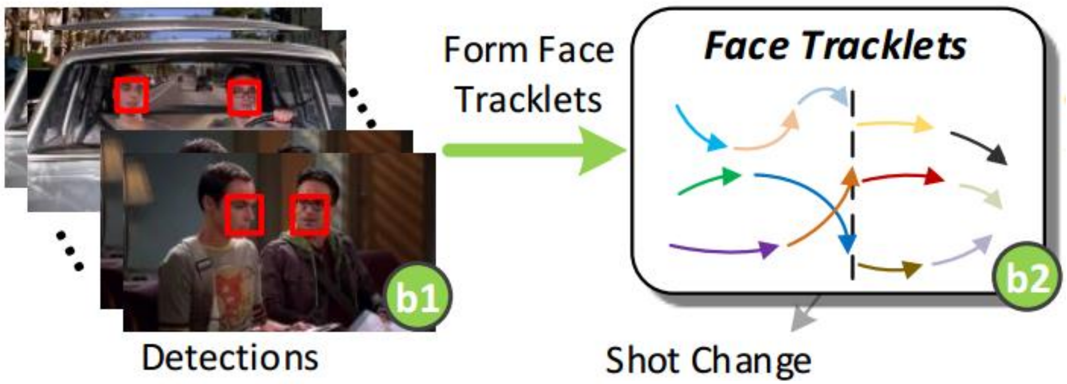
Minah

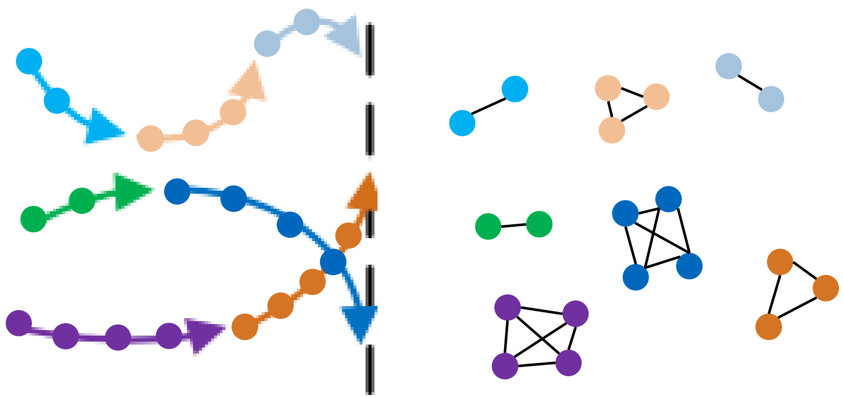
Yura

Hyeri

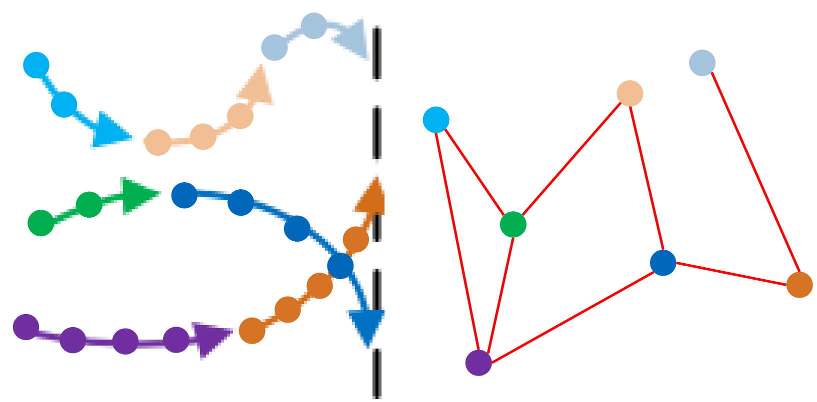
Need **discriminative** features

Discover constraints from videos



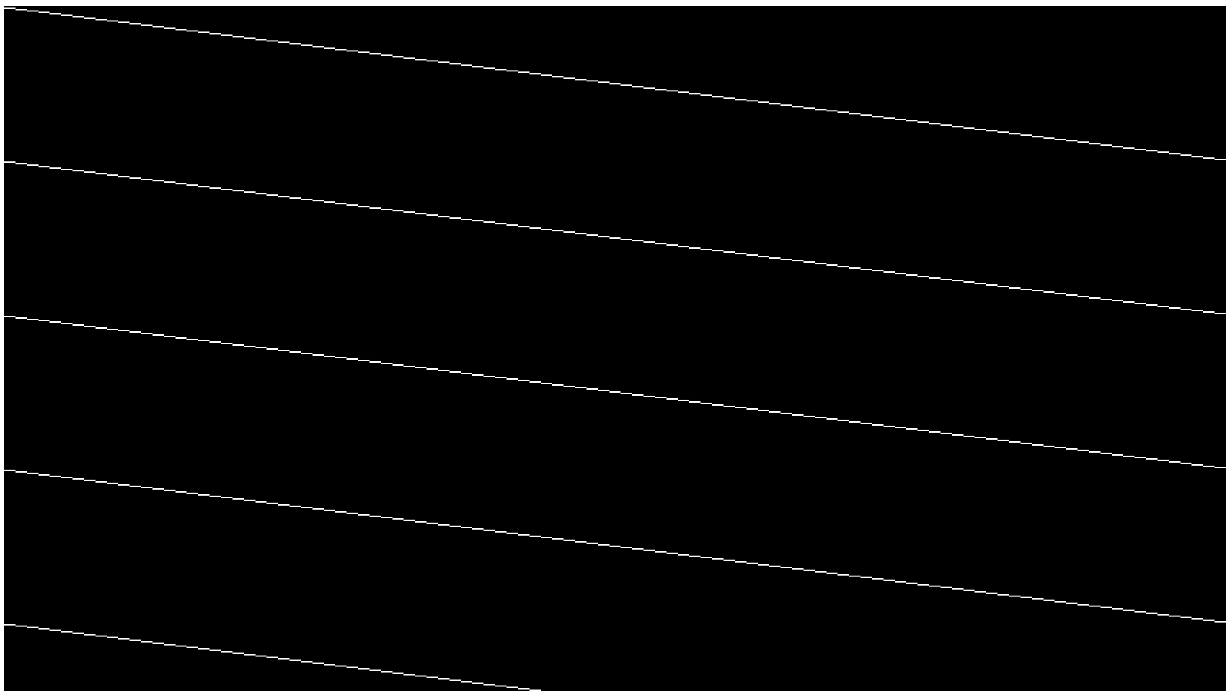


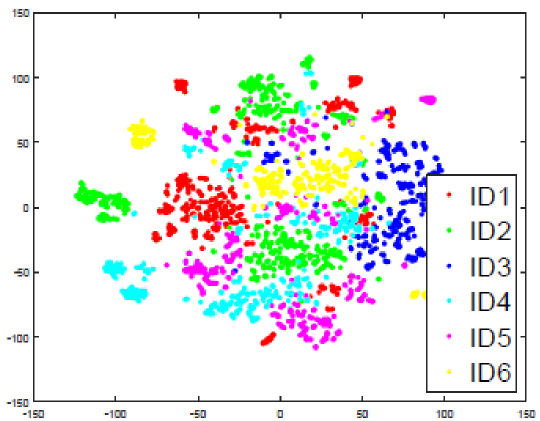
Same person



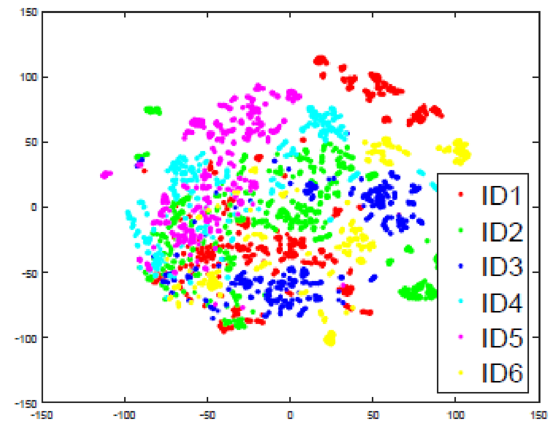
Different persons

Qualitative results

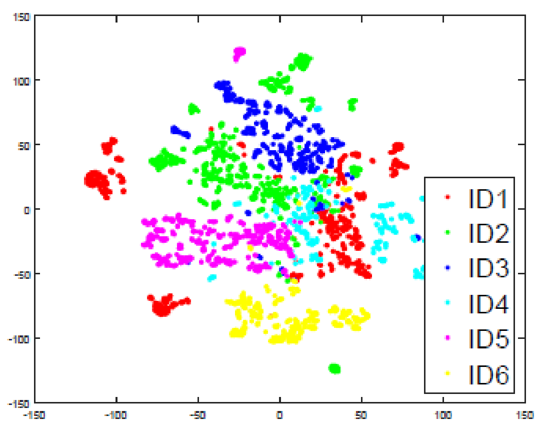




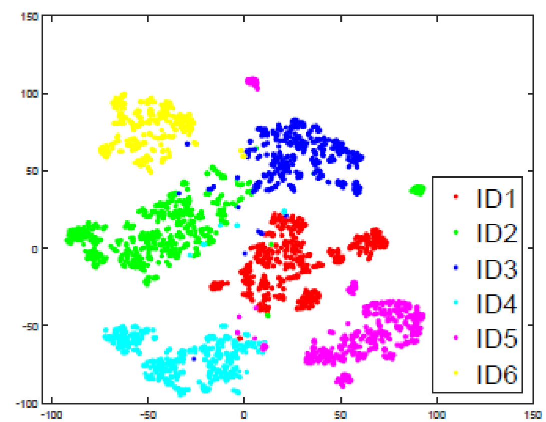
HOG (4356-D)



AlexNet (4096-D)

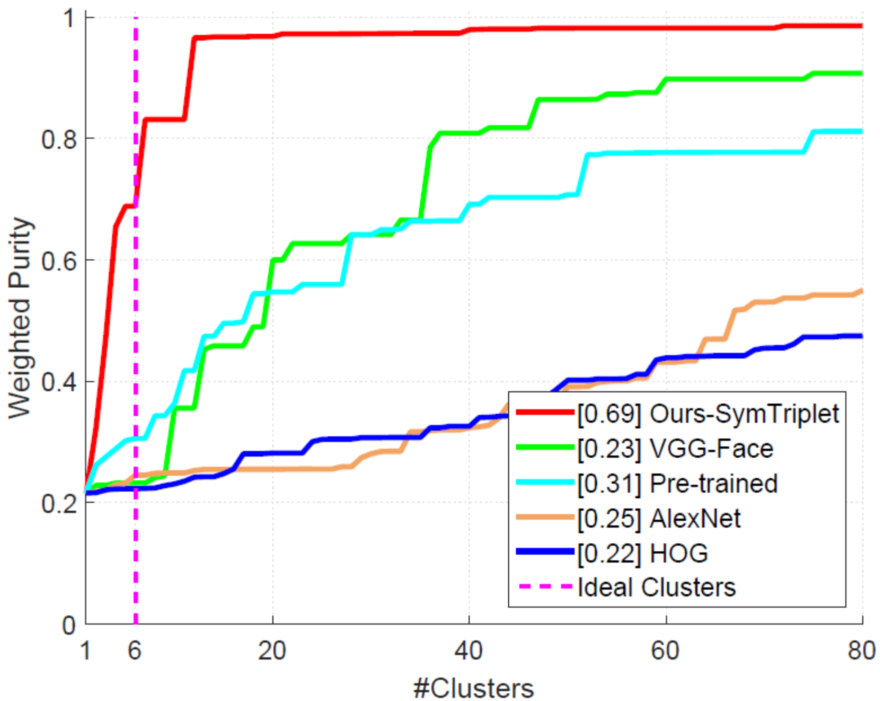


Pre-trained (4096-D)

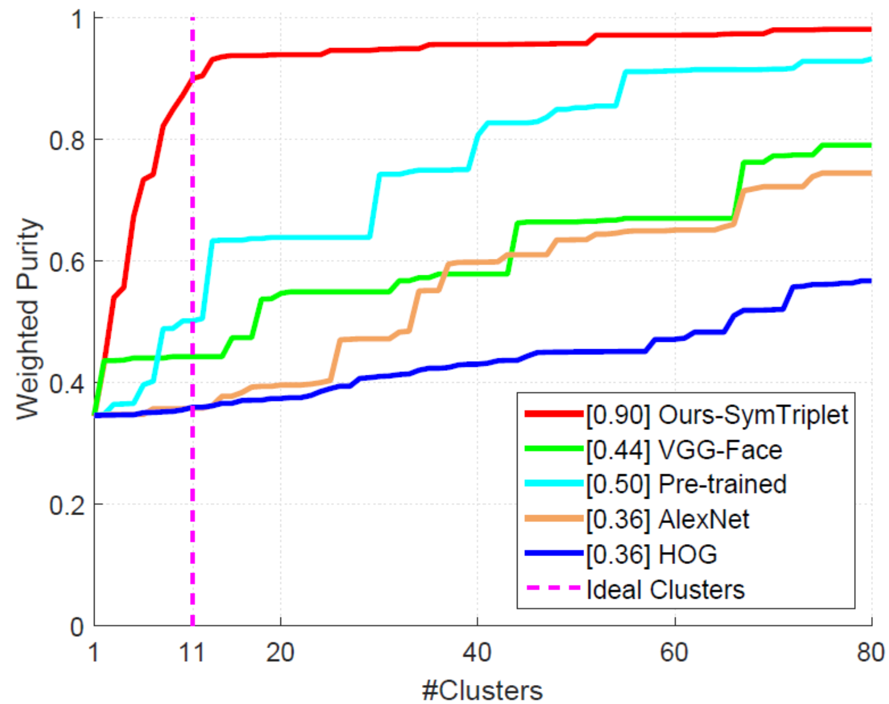


Ours-SymTriplet (64-D)

Quantitative Results



T-ara



Bruno Mars

Things to remember

- Learning distance is crucial for
  - Matching
  - Retrieval
  - Recognition
  - Re-identification
- Two common strategies
  - Siamese network
  - Triplet network