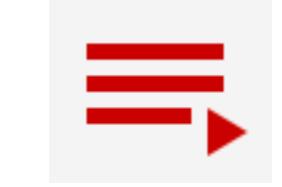




Boulder

# Computer Vision; Image Classification; Transfer Learning



[YouTube Playlist](#)

**Maziar Raissi**

**Assistant Professor**

Department of Applied Mathematics

University of Colorado Boulder

[maziar.raissi@colorado.edu](mailto:maziar.raissi@colorado.edu)

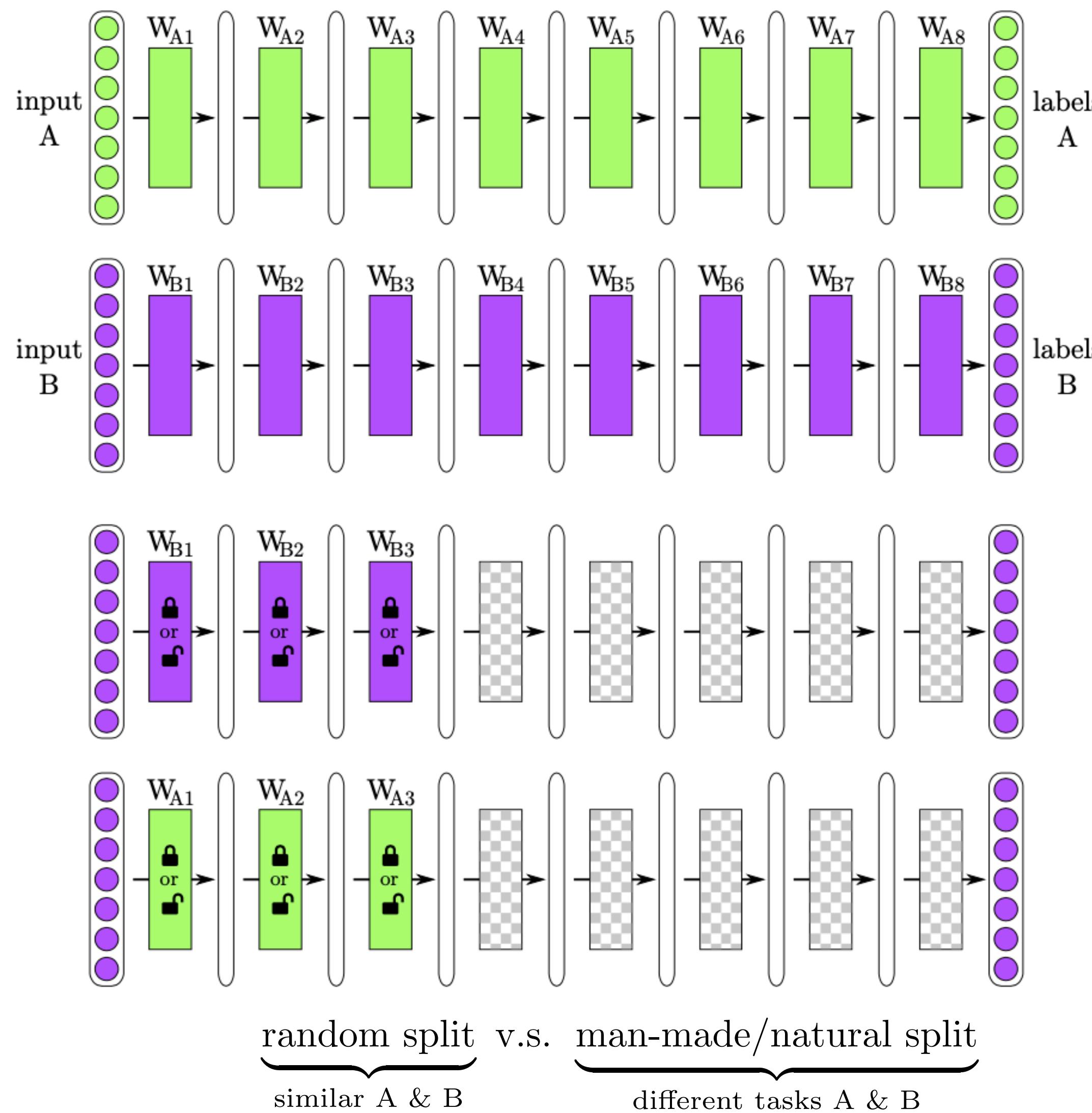


Boulder

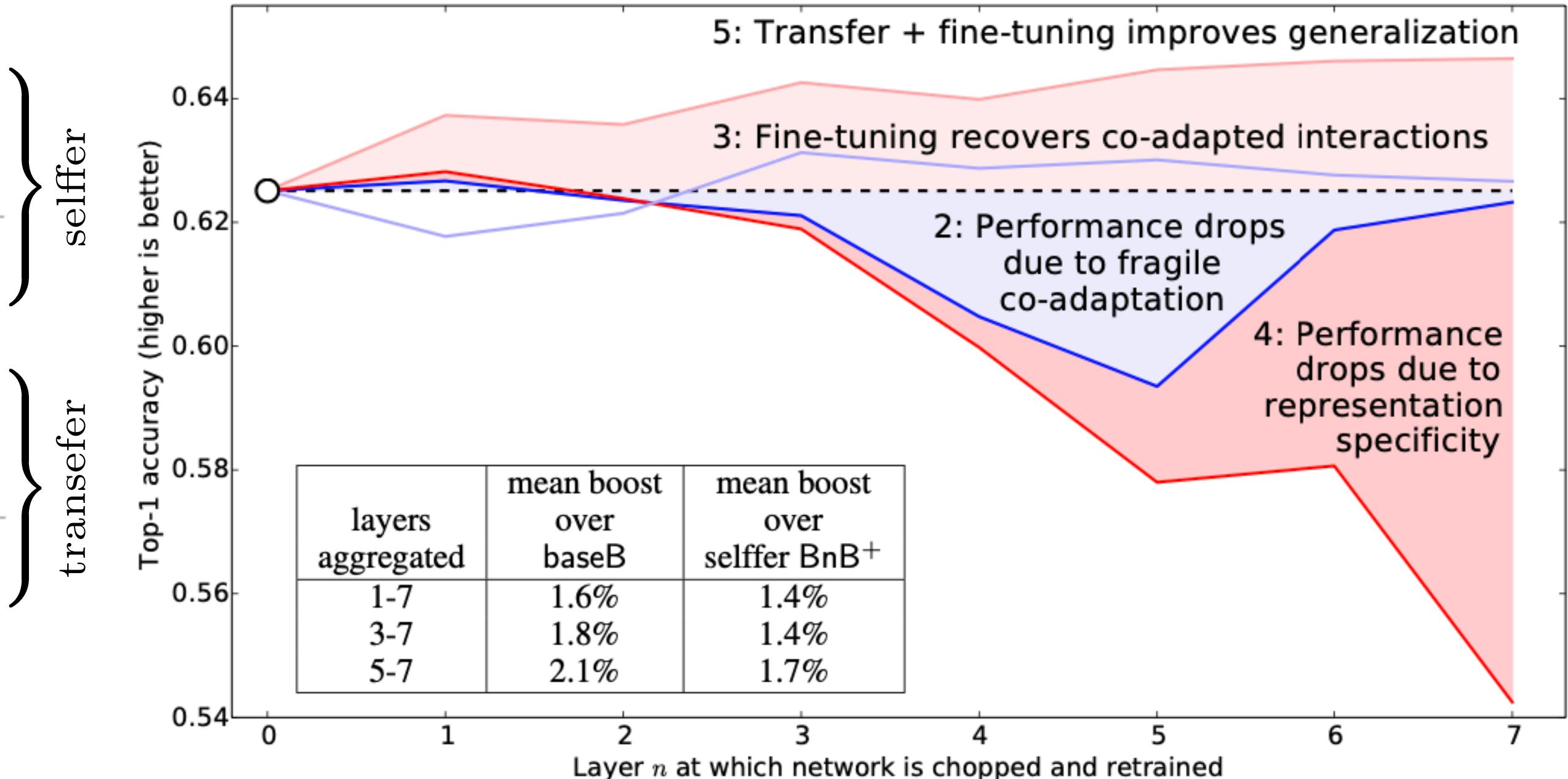
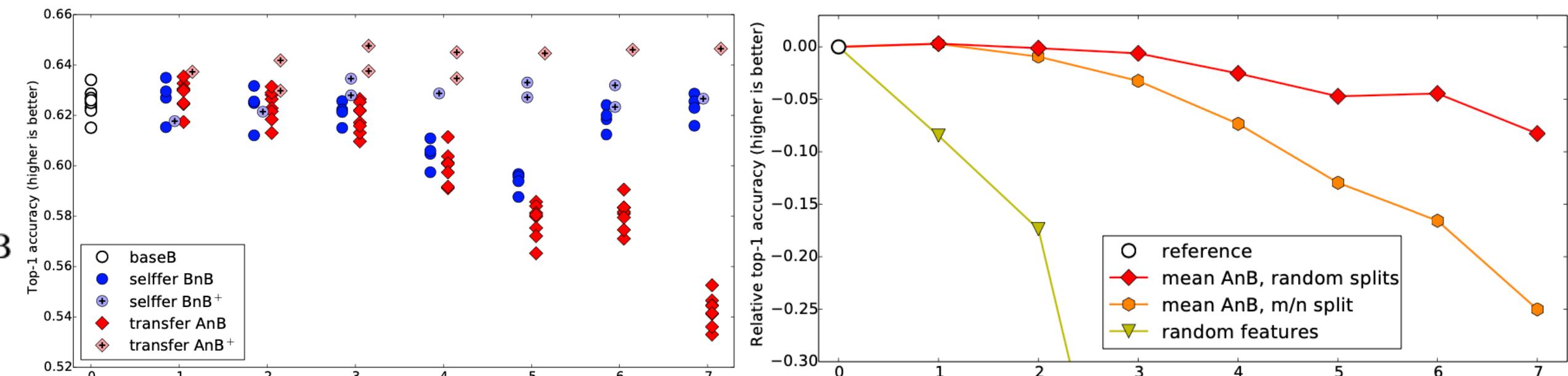


[YouTube Video](#)

# How transferable are features in deep neural networks?



- Can we quantify the degree to which a particular layer is general or specific?
- Does the transition occur suddenly at a single layer, or is it spread out over several layers?
- Where does this transition take place: near the first, middle, or last layer of the network?





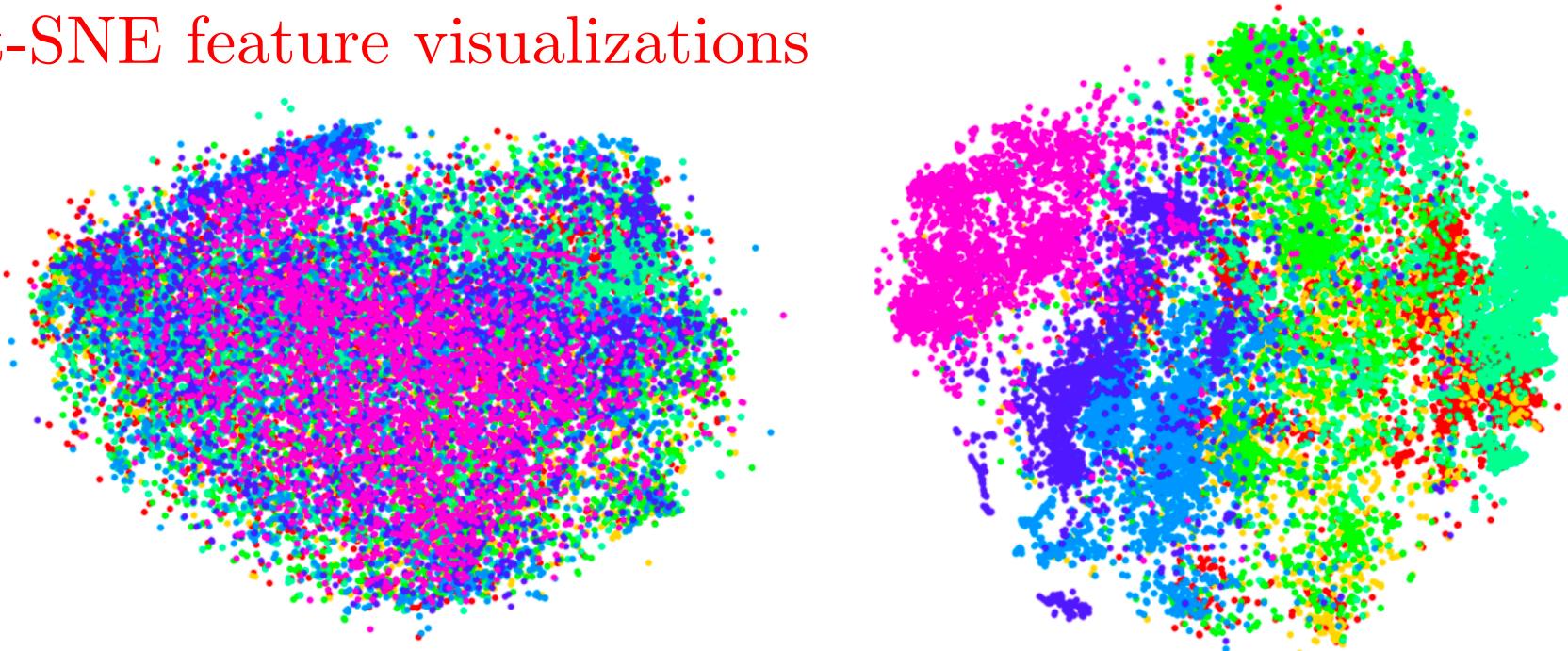
Boulder

# DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition



[YouTube Video](#)

t-SNE feature visualizations

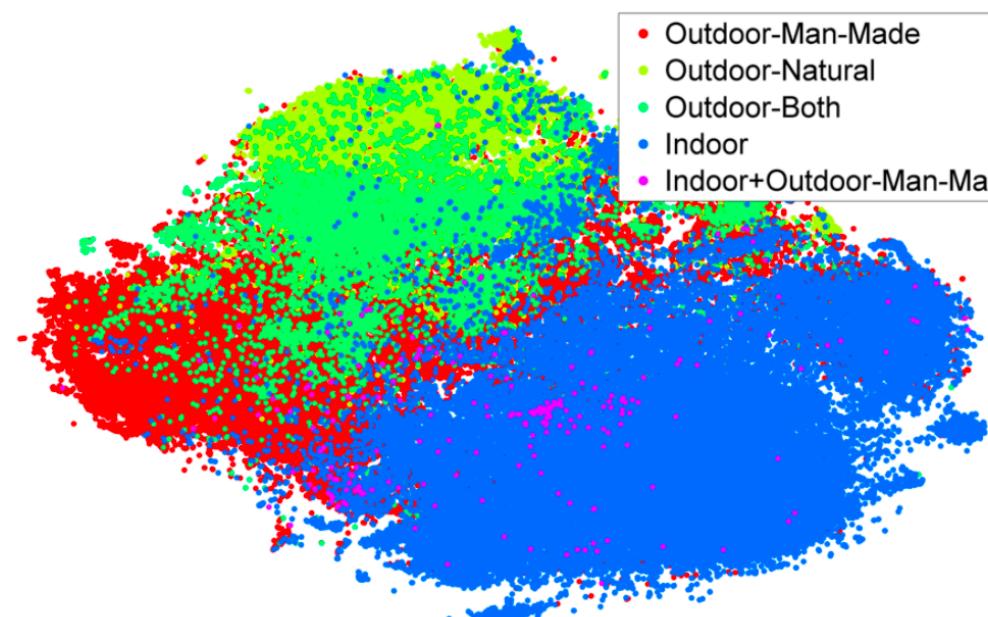


DeCAF<sub>1</sub>  
first pooling layer

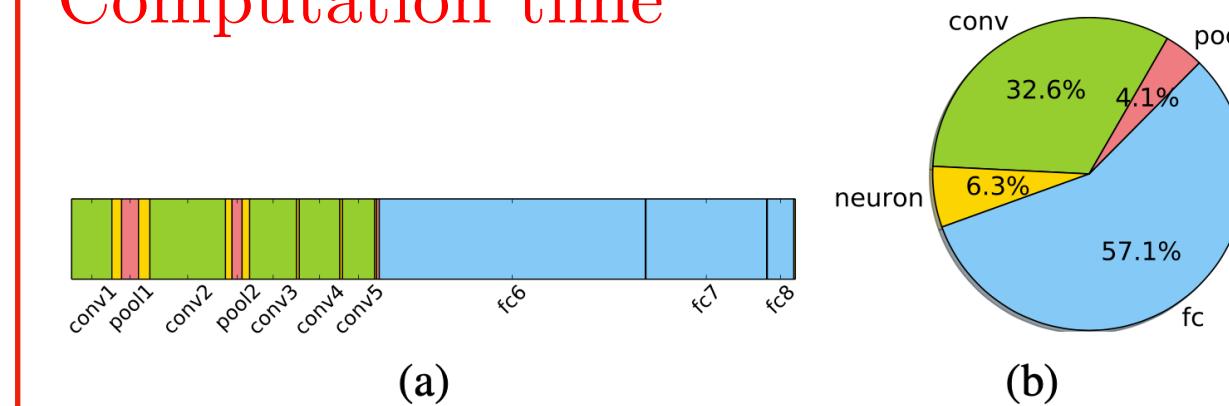
- structure, construction
- covering
- commodity, trade good, good
- conveyance, transport
- invertebrate
- bird
- hunting dog

First layers learn “low-level” features, whereas the latter layers learn semantic or “high-level” features.

Features trained on ILSVRC-2012 generalized to SUN-397

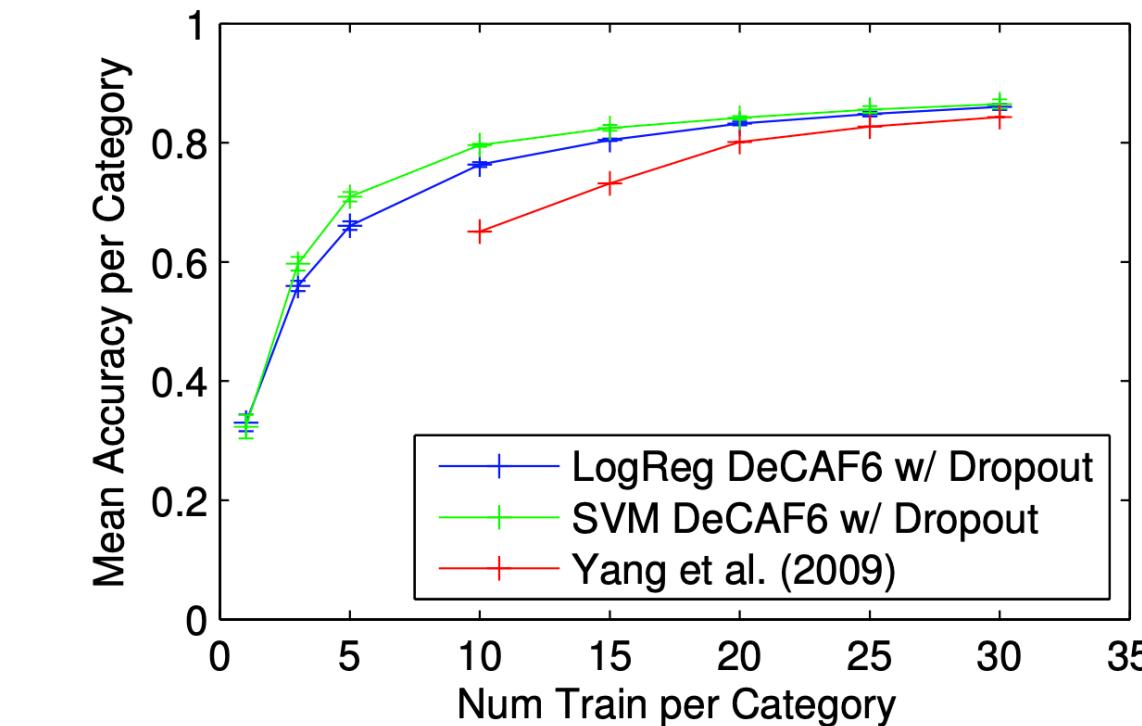


Computation time



Object recognition (Caltech-101)

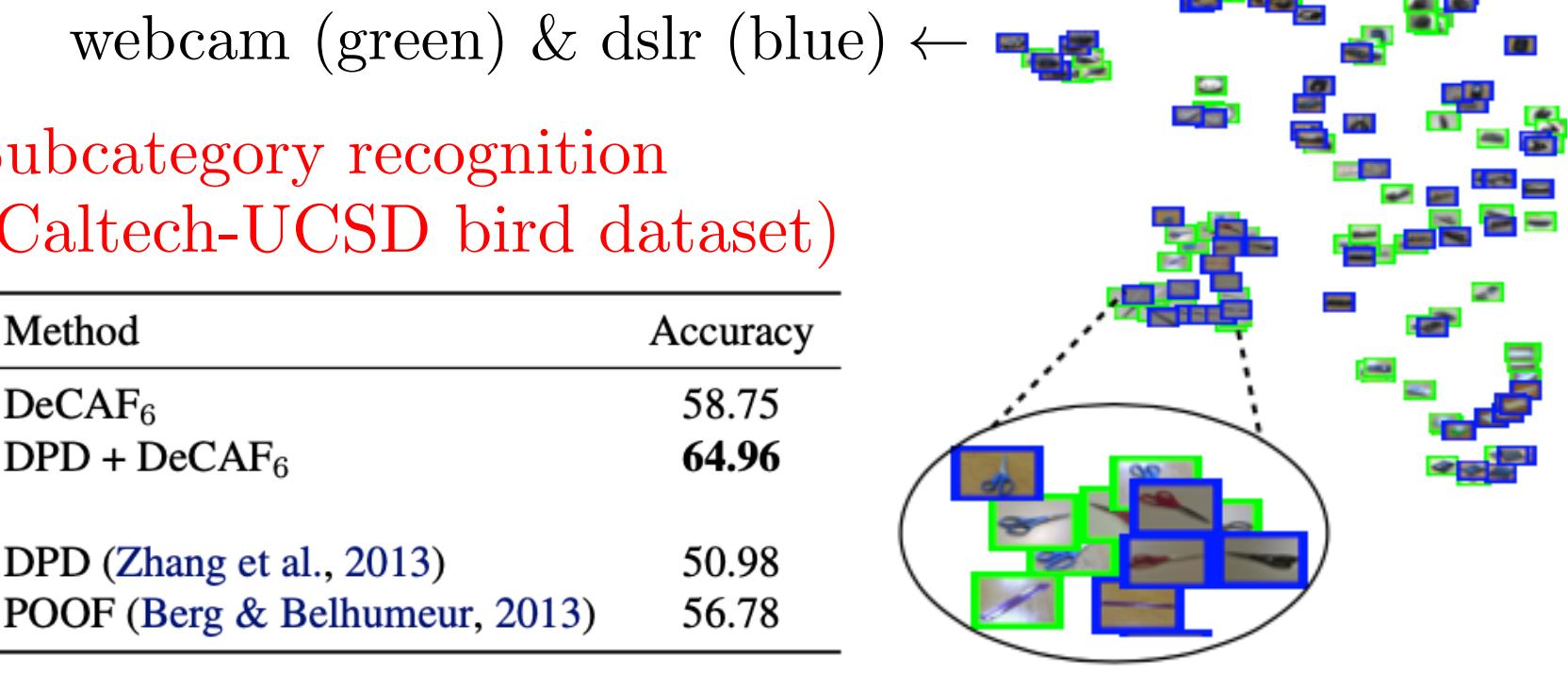
	DeCAF <sub>5</sub>	DeCAF <sub>6</sub>	DeCAF <sub>7</sub>
LogReg	63.29 ± 6.6	84.30 ± 1.6	84.87 ± 0.6
LogReg with Dropout	-	86.08 ± 0.8	85.68 ± 0.6
SVM	77.12 ± 1.1	84.77 ± 1.2	83.24 ± 1.2
SVM with Dropout	-	<b>86.91 ± 0.7</b>	85.51 ± 0.9
Yang et al. (2009)		84.3	
Jarrett et al. (2009)		65.5	



Domain adaptation (Office dataset)

The dataset contains three domains: Amazon, which consists of product images taken from [amazon.com](http://amazon.com); and Webcam and Dslr, which consist of images taken in an office environment using a webcam or digital SLR camera, respectively.

	Amazon → Webcam			Dslr → Webcam		
	SURF	DeCAF <sub>6</sub>	DeCAF <sub>7</sub>	SURF	DeCAF <sub>6</sub>	DeCAF <sub>7</sub>
Logistic Reg. (S)	9.63 ± 1.4	48.58 ± 1.3	53.56 ± 1.5	24.22 ± 1.8	88.77 ± 1.2	87.38 ± 2.2
SVM (S)	11.05 ± 2.3	52.22 ± 1.7	53.90 ± 2.2	38.80 ± 0.7	91.48 ± 1.5	89.15 ± 1.7
Logistic Reg. (T)	24.33 ± 2.1	72.56 ± 2.1	74.19 ± 2.8	24.33 ± 2.1	72.56 ± 2.1	74.19 ± 2.8
SVM (T)	51.05 ± 2.0	78.26 ± 2.6	78.72 ± 2.3	51.05 ± 2.0	78.26 ± 2.6	78.72 ± 2.3
Logistic Reg. (ST)	19.89 ± 1.7	75.30 ± 2.0	76.32 ± 2.0	36.55 ± 2.2	92.88 ± 0.6	91.91 ± 2.0
SVM (ST)	23.19 ± 3.5	80.66 ± 2.3	79.12 ± 2.1	46.32 ± 1.1	<b>94.79 ± 1.2</b>	92.96 ± 2.0
Daume III (2007)	40.26 ± 1.1	<b>82.14 ± 1.9</b>	81.65 ± 2.4	55.07 ± 3.0	91.25 ± 1.1	89.52 ± 2.2
Hoffman et al. (2013)	37.66 ± 2.2	80.06 ± 2.7	80.37 ± 2.0	53.65 ± 3.3	93.25 ± 1.5	91.45 ± 1.5
Gong et al. (2012)	39.80 ± 2.3	75.21 ± 1.2	77.55 ± 1.9	39.12 ± 1.3	88.40 ± 1.0	88.66 ± 1.9
Chopra et al. (2013)		58.85			78.21	



Scene recognition (SUN-397)

	DeCAF <sub>6</sub>	DeCAF <sub>7</sub>
LogReg	<b>40.94 ± 0.3</b>	40.84 ± 0.3
SVM	39.36 ± 0.3	40.66 ± 0.3
Xiao et al. (2010)		38.0

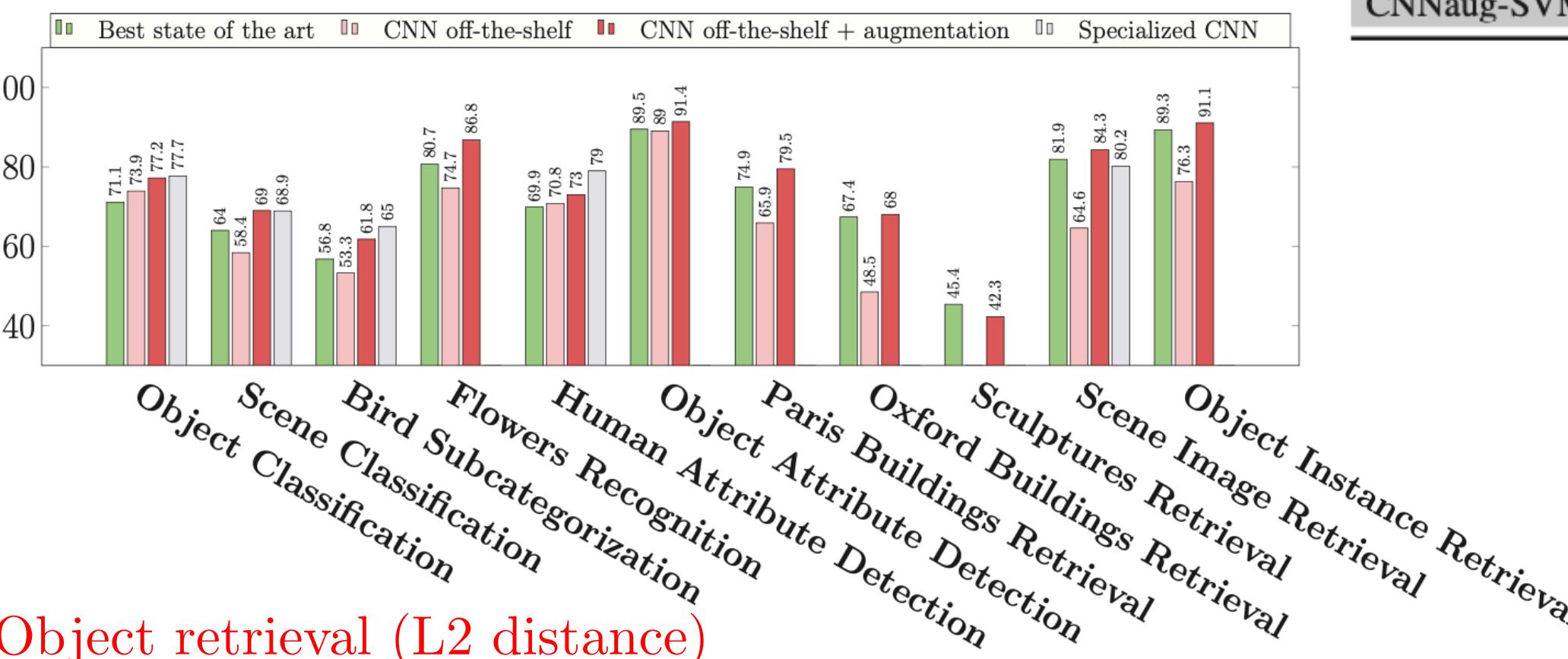
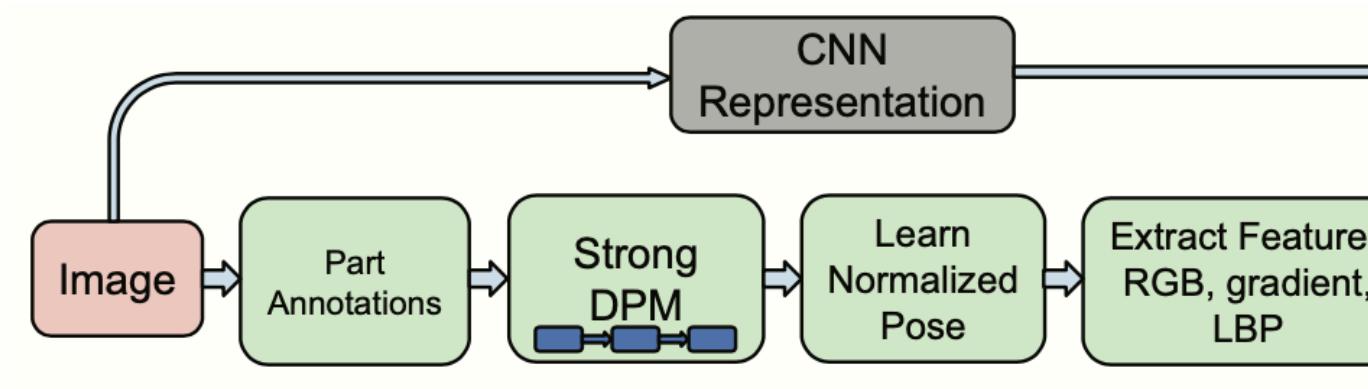


Boulder



[YouTube Video](#)

# CNN Features off-the-shelf: an Astounding Baseline for Recognition



Object retrieval (L2 distance)

	Dim	Oxford5k	Paris6k	Sculp6k	Holidays	UKBench
BoB[3]	N/A	N/A	N/A	<b>45.4</b> [3]	N/A	N/A
BoW	200k	36.4[20]	46.0[35]	8.1[3]	54.0[4]	70.3[20]
IFV[33]	2k	41.8[20]	-	-	62.6[20]	83.8[20]
VLAD[4]	32k	55.5 [4]	-	-	64.6[4]	-
CVLAD[52]	64k	47.8[52]	-	-	81.9[52]	89.3[52]
HE+burst[17]	64k	64.5[42]	-	-	78.0[42]	-
AHE+burst[17]	64k	66.6[42]	-	-	79.4[42]	-
Fine vocab[26]	64k	74.2[26]	74.9[26]	-	74.9[26]	-
ASMK*+MA[42]	64k	80.4[42]	77.0[42]	-	81.0[42]	-
ASMK+MA[42]	64k	<b>81.7</b> [42]	78.2[42]	-	82.2[42]	-
CNN	4k	32.2	49.5	24.1	64.2	76.0
CNN-ss	32-120k	55.6	69.7	31.1	76.9	86.9
CNNaug-ss	4-15k	<b>68.0</b>	<b>79.5</b>	42.3	<b>84.3</b>	<b>91.1</b>
CNN+BOW[16]	2k	-	-	-	<b>80.2</b>	-

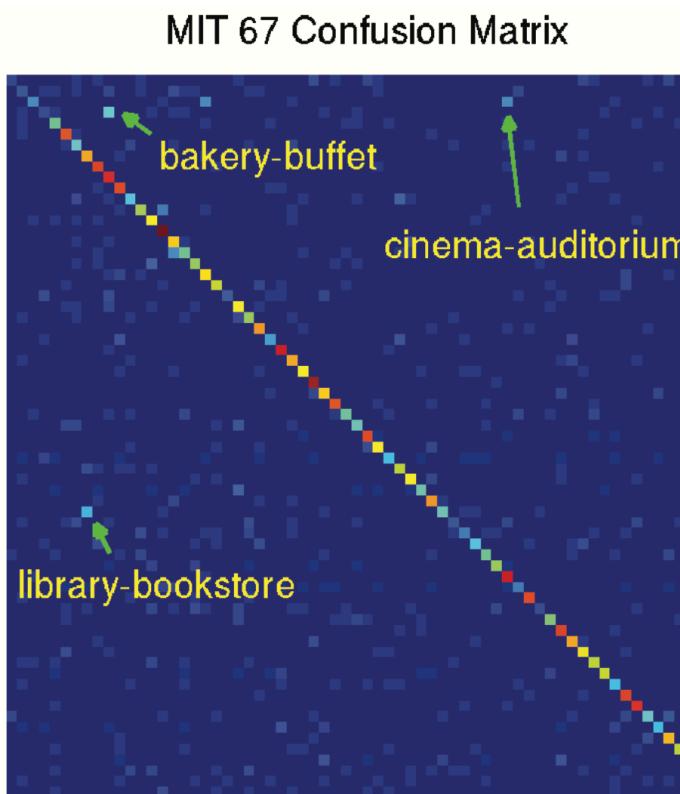
Sharif Razavian, Ali, et al. "CNN features off-the-shelf: an astounding baseline for recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014.

Pascal VOC 2007 Image Classification

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
GHM[8]	76.7	74.7	53.8	72.1	40.4	71.7	83.6	66.5	52.5	57.5	62.8	51.1	81.4	71.5	86.5	36.4	55.3	60.6	80.6	57.8	64.7
AGS[11]	82.2	83.0	58.4	76.1	<b>56.4</b>	<b>77.5</b>	<b>88.8</b>	69.1	<b>62.2</b>	61.8	64.2	51.3	<b>85.4</b>	<b>80.2</b>	91.1	48.1	61.7	<b>67.7</b>	86.3	70.9	71.1
NUS[39]	82.5	79.6	64.8	73.4	54.2	75.0	77.5	79.2	46.2	62.7	41.4	74.6	85.0	76.8	91.1	53.9	61.0	67.5	83.6	70.6	70.5
CNN-SVM	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5	66.5	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.2	71.8	73.9
CNNaug-SVM	<b>90.1</b>	<b>84.4</b>	<b>86.5</b>	<b>84.1</b>	48.4	73.4	86.7	<b>85.4</b>	61.3	<b>67.6</b>	<b>69.6</b>	<b>84.0</b>	<b>85.4</b>	80.0	<b>92.0</b>	<b>56.9</b>	<b>76.7</b>	67.3	<b>89.1</b>	<b>74.9</b>	<b>77.2</b>

MIT-67 indoor scenes dataset

Method	mean Accuracy
ROI + Gist[36]	26.1
DPM[30]	30.4
Object Bank[24]	37.6
RBow[31]	37.9
BoP[21]	46.1
miSVM[25]	46.4
D-Parts[40]	51.4
IFV[21]	60.8
MLrep[9]	64.0
CNN-SVM	58.4
CNNaug-SVM	<b>69.0</b>
CNN(AlexConvNet)+multiscale pooling [16]	68.9



mean accuracy: mean of the confusion matrix diagonal

CUB 200-2011 Bird dataset

Method	Part info	mean Accuracy
Sift+Color+SVM[45]	✗	17.3
Pose pooling kernel[49]	✓	28.2
RF[47]	✓	19.2
DPD[50]	✓	51.0
Poof[5]	✓	56.8
CNN-SVM	✗	53.3
CNNaug-SVM	✗	<b>61.8</b>
DPD+CNN(DeCaf)+LogReg[10]	✓	<b>65.0</b>

H3D Human Attributes dataset

Method	male	lg hair	glasses	hat	tshirt	lg slvs	shorts	jeans	lg pants	mAP
Freq[6]	59.3	30.0	22.0	16.6	23.5	49.0	17.9	33.8	74.7	36.3
SPM[6]	68.1	40.0	25.9	35.3	30.6	58.0	31.4	39.5	84.3	45.9
Poselets[6]	82.4	<b>72.5</b>	<b>55.6</b>	60.1	51.2	74.2	45.5	54.7	90.3	65.2
DPD[50]	83.7	70.0	38.1	<b>73.4</b>	49.8	78.1	64.1	<b>78.1</b>	93.5	69.9
CNN-SVM	83.0	67.6	39.7	66.8	52.6	82.2	78.2	71.7	95.2	70.8
CNNaug-SVM	<b>84.8</b>	71.0	42.5	66.9	<b>57.7</b>	<b>84.0</b>	<b>79.1</b>	75.7	<b>95.3</b>	<b>73.0</b>

CNNaug: augment the training set by adding cropped and rotated samples



Boulder

# Return of the Devil in the Details: Delving Deep into Convolutional Nets

$I \rightarrow \text{image}$

$\phi \rightarrow \text{encoding function}$

$\phi(I) \in \mathbb{R}^d \rightarrow \text{vector image representation}$

## Shallow Representation (IFV)

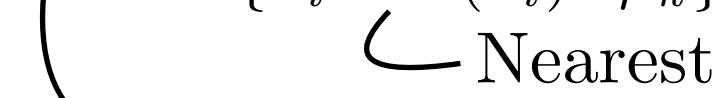
Improved Fisher Vector (IFV)

- Extract a dense collection of patches and corresponding local descriptors  $x_i \in \mathbb{R}^D$

(e.g., SIFT) from the image at multiple scales

- Each descriptor  $x_i$  is then soft-quantized (i.e., clustered) using a Gaussian Mixture Model with K components.

$$u_k := \sum_{\{x_i: NN(x_i)=\mu_k\}} (x_i - \mu_k) \in \mathbb{R}^D$$

 Nearest Neighbor

accumulated first order differences

$$v_k := \sum_{\{x_i: NN(x_i)=\mu_k\}} (x_i - \mu_k)^2 \in \mathbb{R}^D$$

 accumulated second order differences

$$\phi_{\text{FV}}(I) = [u_1; v_1; \dots; u_K; v_K] \in \mathbb{R}^{2KD}$$

$$\phi_{\text{IFV}}(I) \leftarrow \phi_{\text{FV}}(I) \triangleright \text{sign}(\cdot) \sqrt{|\cdot|} \triangleright \ell_2 \text{ normalize}$$

## Deep representation (CNN) with pre-training

$\phi_{\text{CNN}}(I) \rightarrow \text{vector activities of penultimate layer}$

**Deep representation (CNN) with pre-training and fine-tuning**

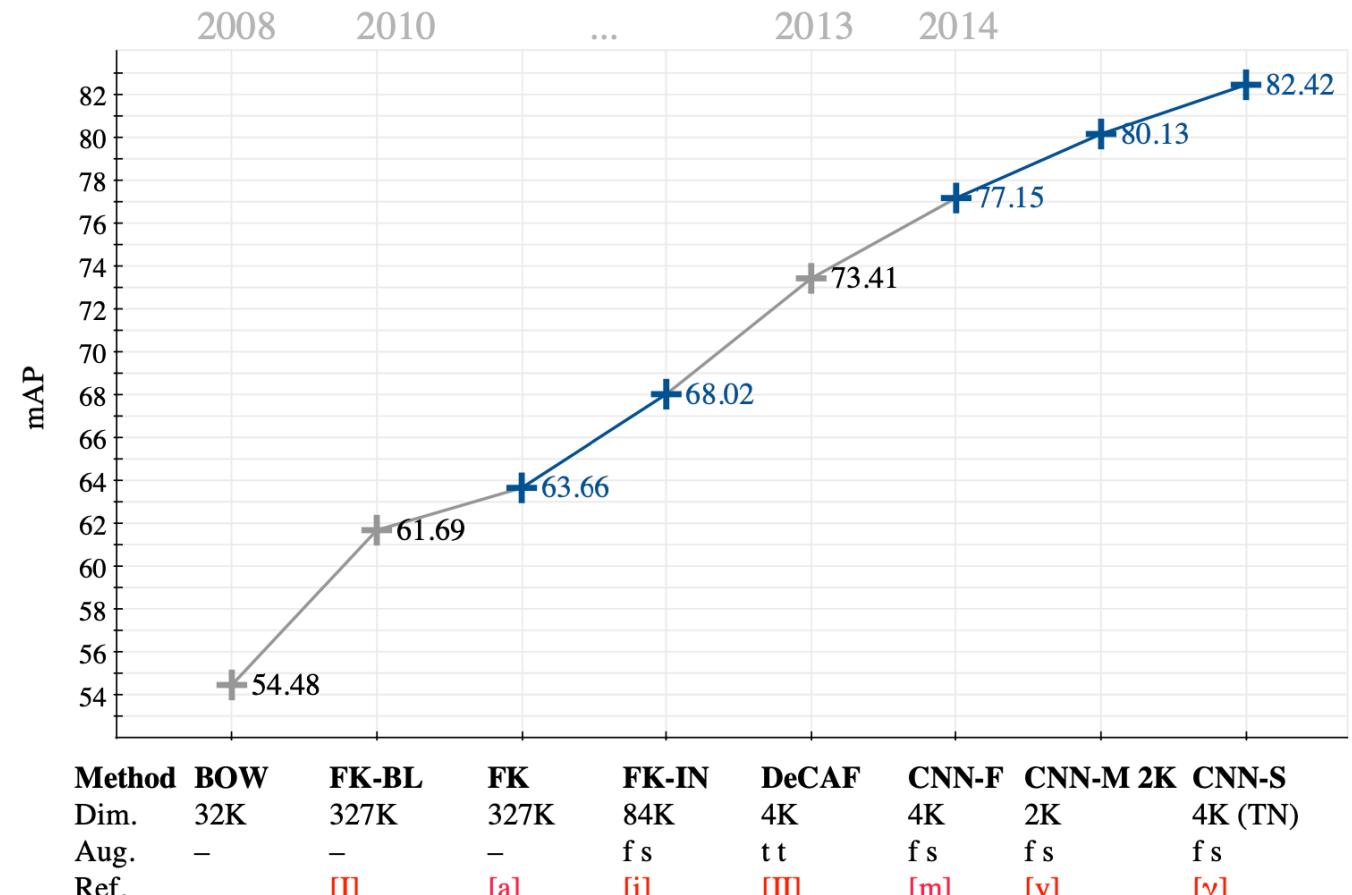
	ILSVRC-2012 (top-5 error)	VOC-2007 (mAP)	VOC-2012 (mAP)	Caltech-101 (accuracy)	Caltech-256 (accuracy)	Method	SPool	Image Aug.	Dim	mAP
(a) FK IN 512	-	68.0	-	-	-	(I) FK BL	spm	-	327K	<b>61.69</b>
(b) CNN F (Fast)	16.7	77.4	79.9	-	-	(II) DECAF	-	(C) t t	327K	<b>73.41</b>
(c) CNN M (Medium)	13.7	79.9	82.5	$87.15 \pm 0.80$	$77.03 \pm 0.46$	(a) FK	spm	-	327K	<b>63.66</b>
(d) CNN M 2048	13.5	80.1	82.4	$86.64 \pm 0.53$	$76.88 \pm 0.35$	(b) FK IN (Intra-normalisation)	spm	-	327K	<b>64.18</b>
(e) CNN S (Slow)	<b>13.1</b>	79.7	82.9	$87.76 \pm 0.66$	<b>77.61 \pm 0.12</b>	(c) FK	(x,y)	-	42K	<b>63.51</b>
(f) CNN S TUNE-CLS	<b>13.1</b>	-	83.0	<b>88.35 \pm 0.56</b>	$77.33 \pm 0.56$	(d) FK IN	(x,y)	-	42K	<b>64.36</b>
(g) CNN S TUNE-RNK	<b>13.1</b>	<b>82.4</b>	<b>83.2</b>	-	-	(e) FK IN	(x,y)	(F) f -	42K	<b>64.35</b>
(h) Zeiler & Fergus [19]	16.1	-	79.0	$86.5 \pm 0.5$	$74.2 \pm 0.3$	(f) FK IN	(x,y)	(C) f s	42K	<b>67.17</b>
(i) Razavian <i>et al.</i> [9], [10]	14.7	77.2	-	-	-	(g) FK IN	(x,y)	(C) s s	42K	<b>66.68</b>
(j) Oquab <i>et al.</i> [8]	18	77.7	78.7 (82.8*)	-	-	(h) FK IN 512	(x,y)	-	84K	<b>65.36</b>
(k) Oquab <i>et al.</i> [16]	-	-	<b>86.3*</b>	-	-	(i) FK IN 512	(x,y)	(C) f s	84K	<b>68.02</b>
(l) Wei <i>et al.</i> [17]	-	81.5 (85.2*)	81.7 (90.3*)	-	-	(j) FK IN COL 512	-	-	82K	<b>52.18</b>
(m) He <i>et al.</i> [29]	13.6	80.1	-	$91.4 \pm 0.7$	-	(k) FK IN 512 COL+	(x,y)	-	166K	<b>66.37</b>
						(l) FK IN 512 COL+	(x,y)	(C) f s	166K	<b>67.93</b>
						(m) CNN F	-	(C) f s	4K	<b>77.38</b>
						(n) CNN S	-	(C) f s	4K	<b>79.74</b>

## Pascal VOC

multi-label dataset

$I_{\text{pos}}, I_{\text{neg}} \rightarrow \text{positive, negative images for each class } c$   
one-vs-rest classification hing loss

$w_c^T \phi(I_{\text{pos}}) > 1 - \xi, w_c^T \phi(I_{\text{pos}}) > -1 + \xi \rightarrow \text{slack}$   
ranking hing loss  $w_c^T \phi(I_{\text{pos}}) > w_c^T \phi(I_{\text{neg}}) + 1 - \xi$



Performance of shallow representations can be significantly improved by adopting **data augmentation**, typically used in deep learning. In spite of this improvement, deep architectures still outperform the shallow methods by a large margin.

(x) CNN M 2048	-	(C) f s	2K	<b>80.10</b>
(y) CNN M 1024	-	(C) f s	1K	<b>79.91</b>
(z) CNN M 128	-	(C) f s	128	<b>78.60</b>
(α) FK+CNN F	(x,y)	(C) f s	88K	<b>77.95</b>
(β) FK+CNN M 2048	(x,y)	(C) f s	86K	<b>80.14</b>
(γ) CNN S TUNE-RNK	-	(C) f s	4K	<b>82.42</b>



Boulder



# Questions?

[YouTube Playlist](#)

---