Domain adaptation and deep learning for large scale object recognition and detection

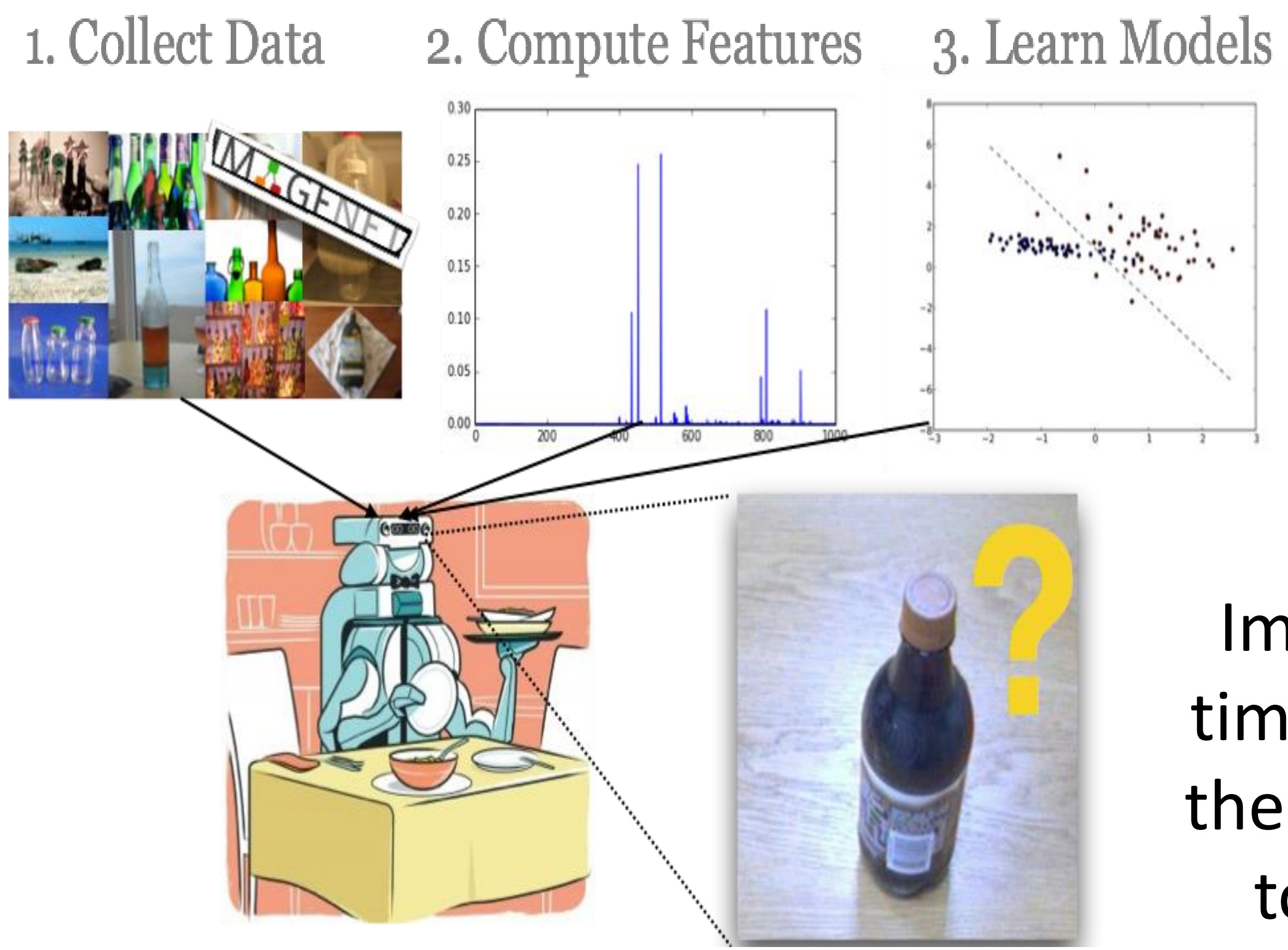
Prof. Trevor Darrell (UCB)

with

Judy Hoffman, Eric Tzeng, Jeff Donahue, Ross Girshick (UCB)

Kate Saenko (UML)

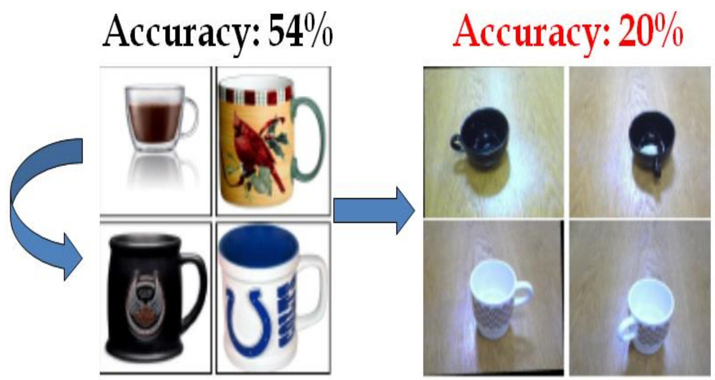
Practical image classification



Images at test time differ from the images used to learn the model!

The Adaptation Problem

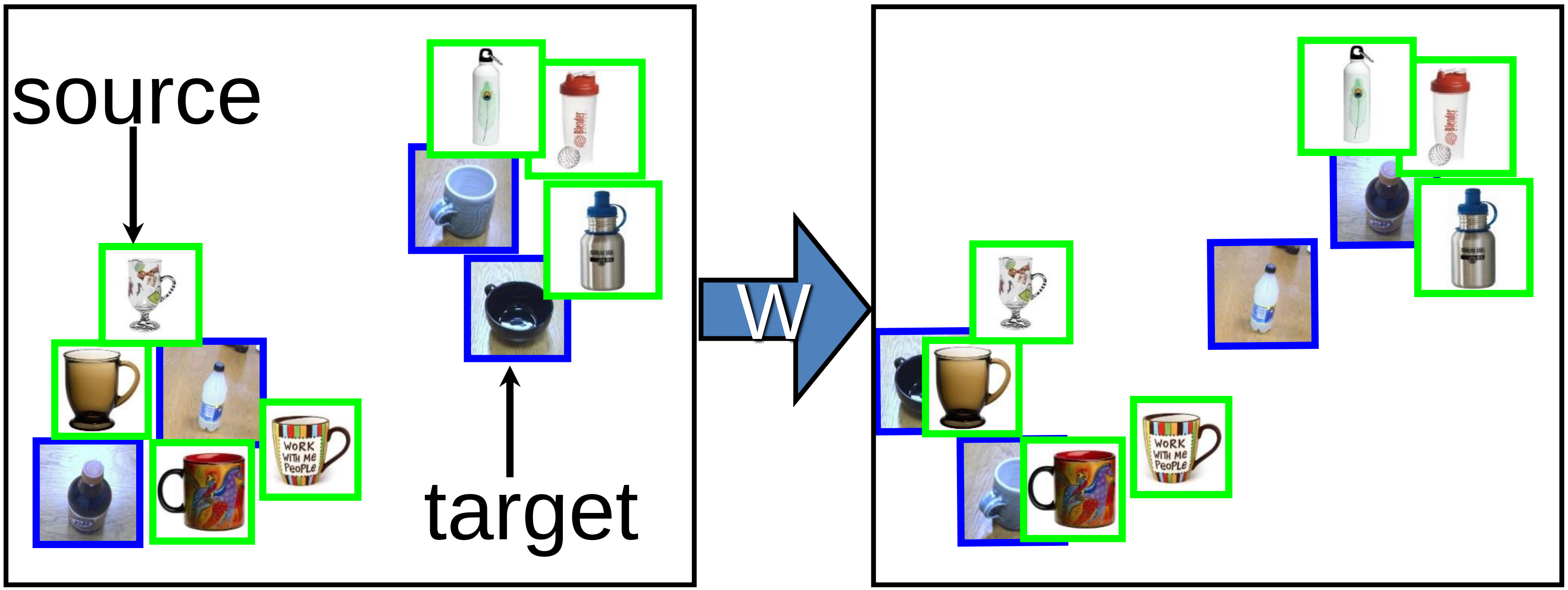
- Images from different visual domains have different appearances
- These differences are what we refer to as “domain shift”
- Given an abundance of data in a source domain (e.g. Amazon), how can we perform a task in a new target domain with very little training data (e.g. Webcam)?



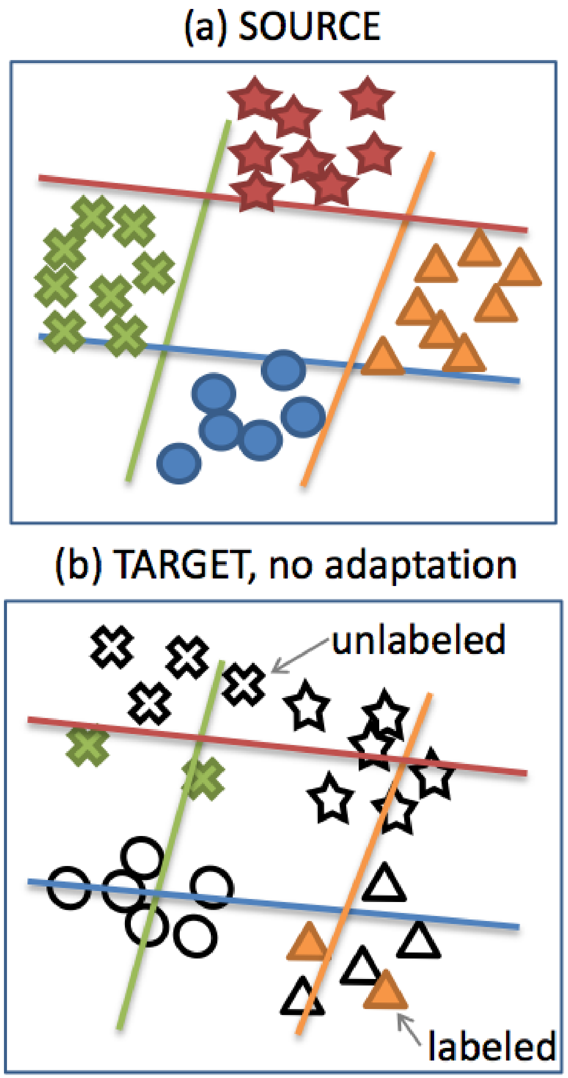
amazon.com

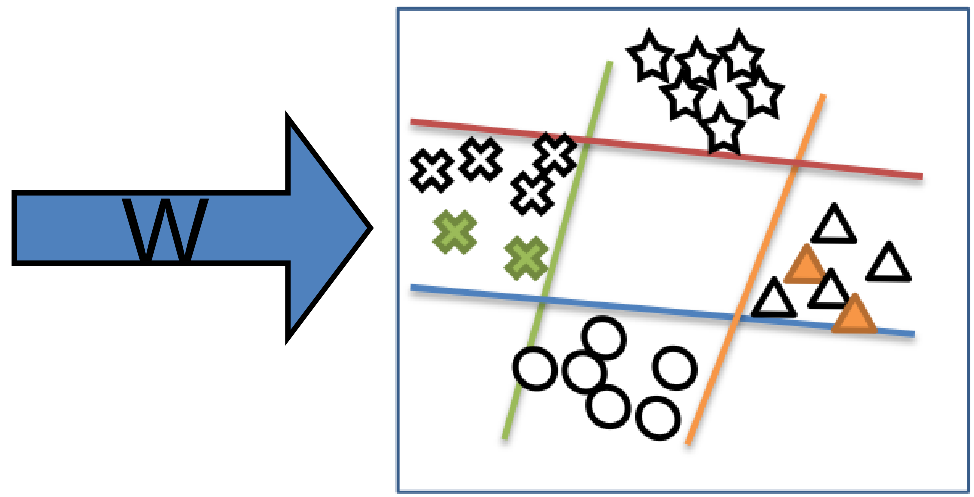
Webcam/DSLR

Background: Feature-space Transformations



Background: Joint feature and Parameter Adaptation

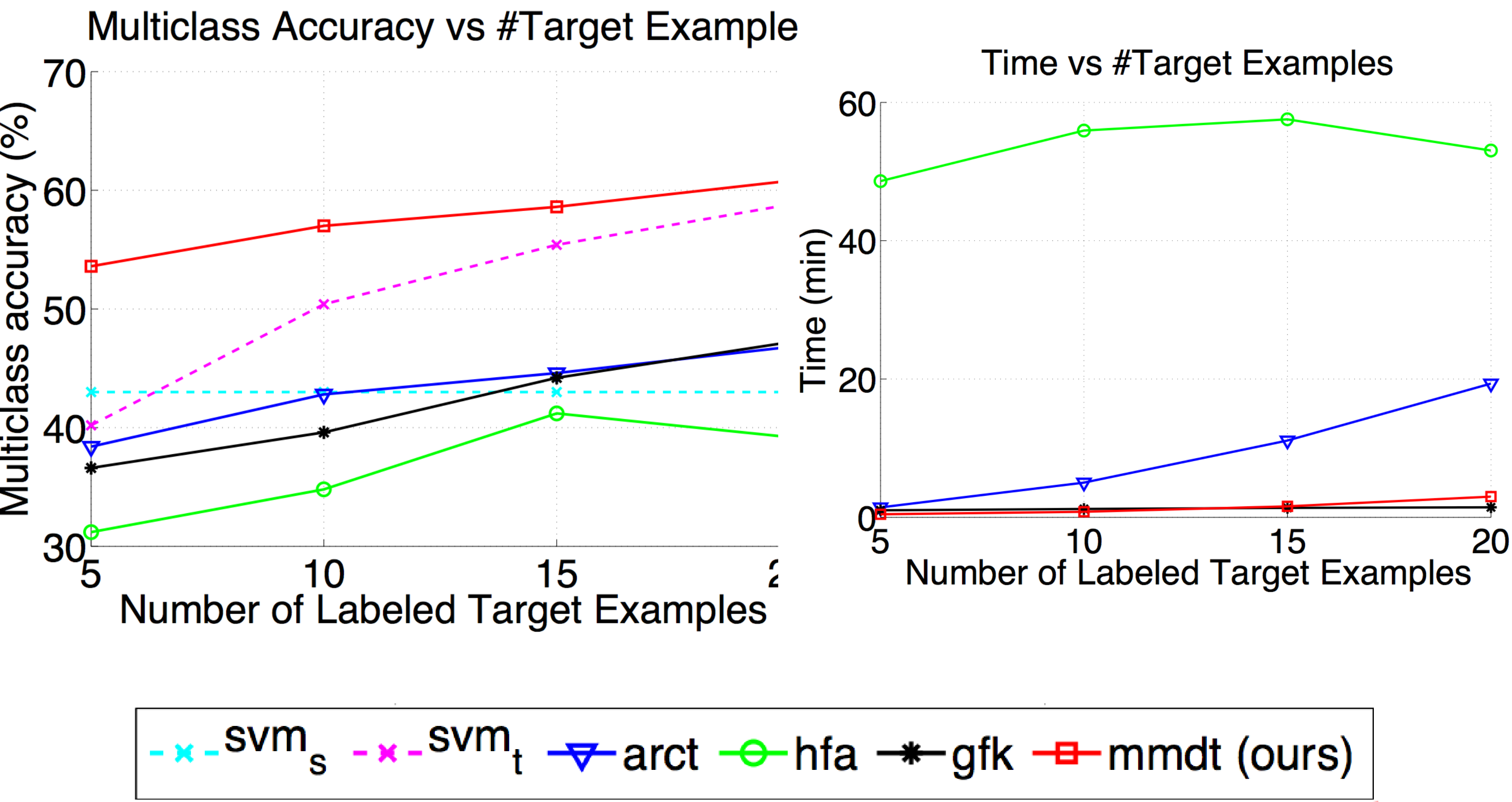




$$\min_{\boldsymbol{W},\boldsymbol{\Theta}} \underbrace{\underbrace{\frac{1}{2}\|\boldsymbol{W}\|_F^2}_{\text{transformation}} + \underbrace{\frac{1}{2}\|\boldsymbol{\Theta}\|_F^2}_{\text{parameter}}}_{\text{regularizer}} + \underbrace{\underbrace{\lambda\,\mathcal{L}(\boldsymbol{W},\boldsymbol{\Theta},\mathbf{Z},\boldsymbol{h})}_{\text{parameter vs target}} + \underbrace{\lambda_{\mathcal{X}}\,\mathcal{L}(\boldsymbol{\Theta},\mathbf{X},\boldsymbol{y})}_{\text{parameter vs source}}}_{\text{loss}}$$

(Hoffman et al., ICLR 13)

Conventional Performance is limited by visual features



Outline

- What does deep learning offer domain adaptation and vice-versa?
- A domain-adaptation perspective on large-scale ImageNet detection
- Beyond binary domain adaptation

Outline

- **What does deep learning offer domain adaptation and vice-versa?**
- A domain-adaptation perspective on large-scale ImageNet detection
- Beyond binary domain adaptation

First attempt: unsupervised deep learning for DA?

- Let’s try a simple idea in the spirit of [1]

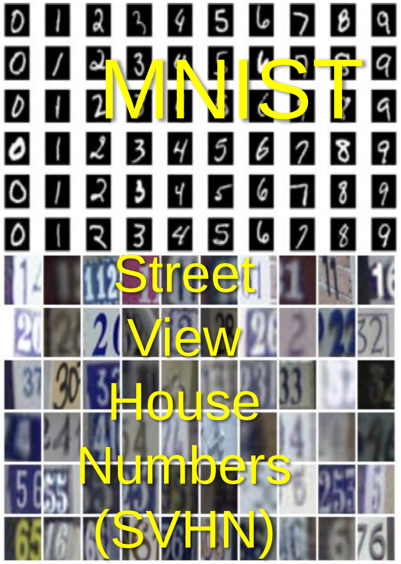
  1) Pre-train unsupervised on both domains to model the data

  2) Then, supervised backprop on the source domain to learn the labels

- But, the pre-training doesn’t seem to help:

  Worse performance for MNIST->SVHN

  Negligibly improved performance for SVHN->MNIST



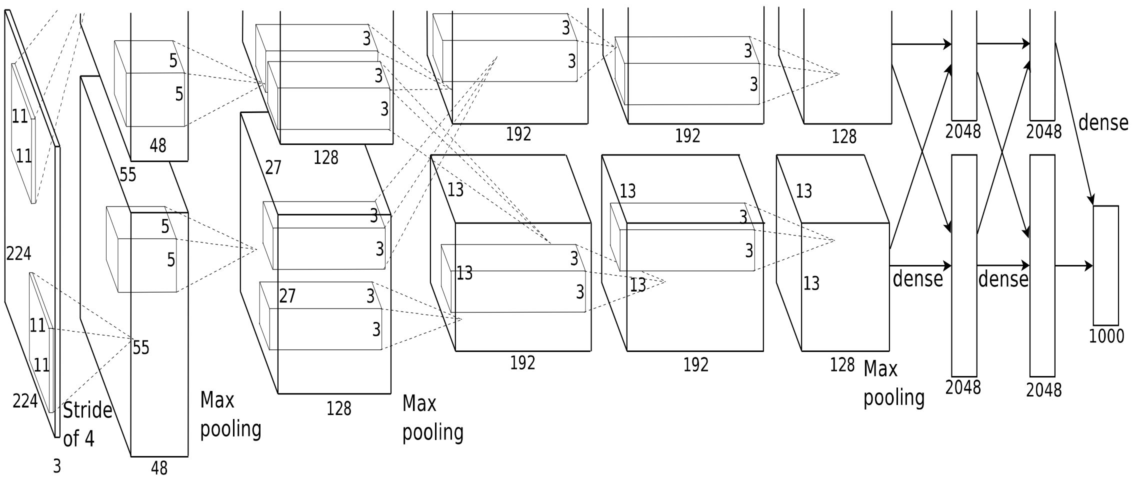
Method	Source	Error Rate	Target	Error Rate
Ours (pre-trained)	MNIST	2.79%	SVHN	66.79%
Baseline (no pre-training)	MNIST	1.56%	SVHN	59.20%
Ours (pre-trained)	SVHN	19.33%	MNIST	41.98%
Baseline (no pre-training)	SVHN	12.38%	MNIST	42.63%

Still has significant domain shift despite cross domain pre-training …

[1] Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. *Science* (2006).

What about supervised CNNs?

- CNNs trained on large amounts of data yield state of the art performance on classification tasks; c.f. AlexNet:



- **Does using a representation learned on large amounts of data remove domain shift?**

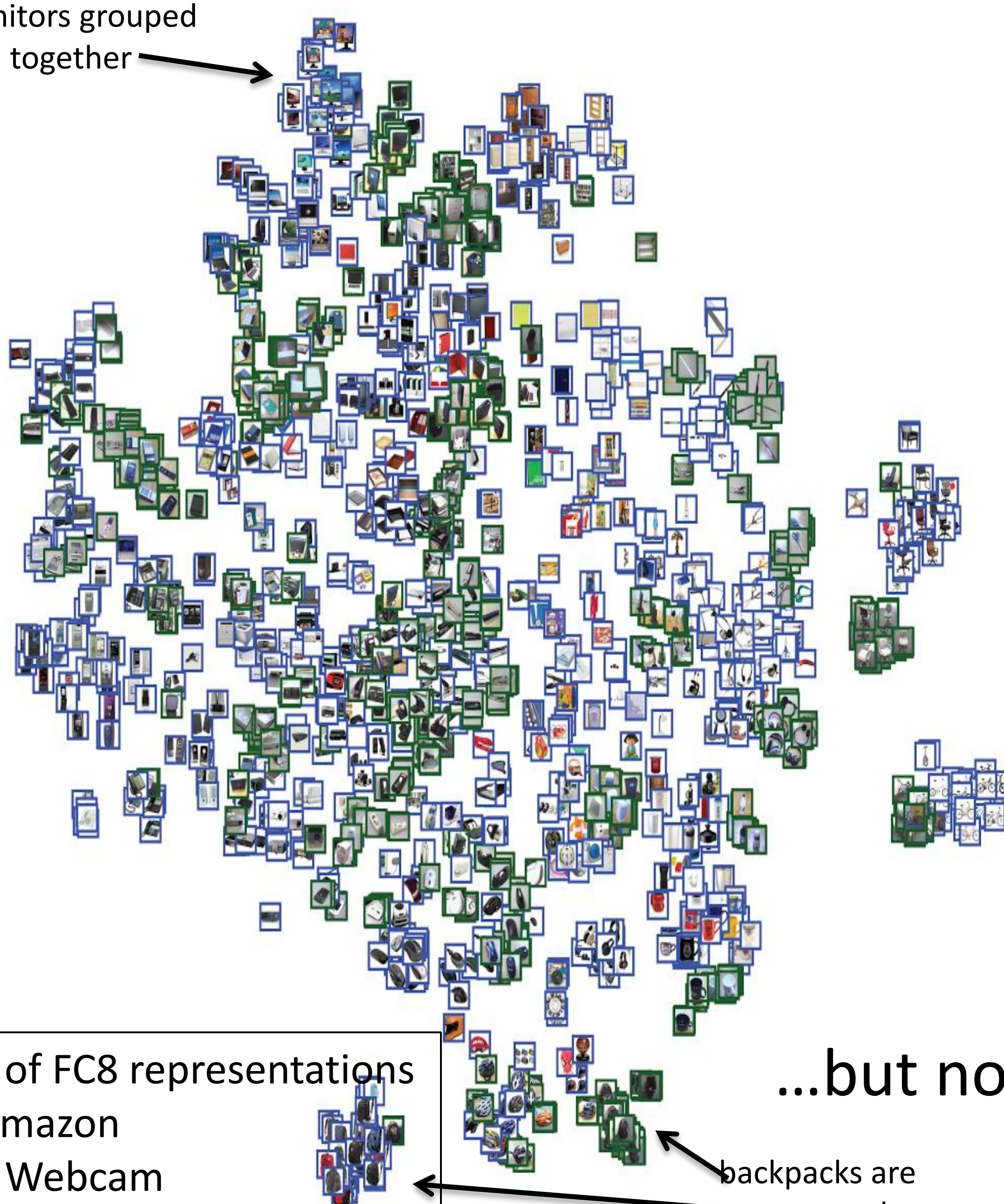
“DeCAF” baselines

	Amazon → Webcam		
	SURF	DeCAF_6	DeCAF_7
Logistic Reg. (S)	9.63 ± 1.4	48.58 ± 1.3	53.56 ± 1.5
SVM (S)	11.05 ± 2.3	52.22 ± 1.7	53.90 ± 2.2
Logistic Reg. (T)	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
Logistic Reg. (ST)	19.89 ± 1.7	75.30 ± 2.0	76.32 ± 2.0
SVM (ST)	23.19 ± 3.5	80.66 ± 2.3	79.12 ± 2.1

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. ICML 2014

monitors grouped together

...kind of!



T-SNE embedding of FC8 representations

Blue = Amazon

Green = Webcam

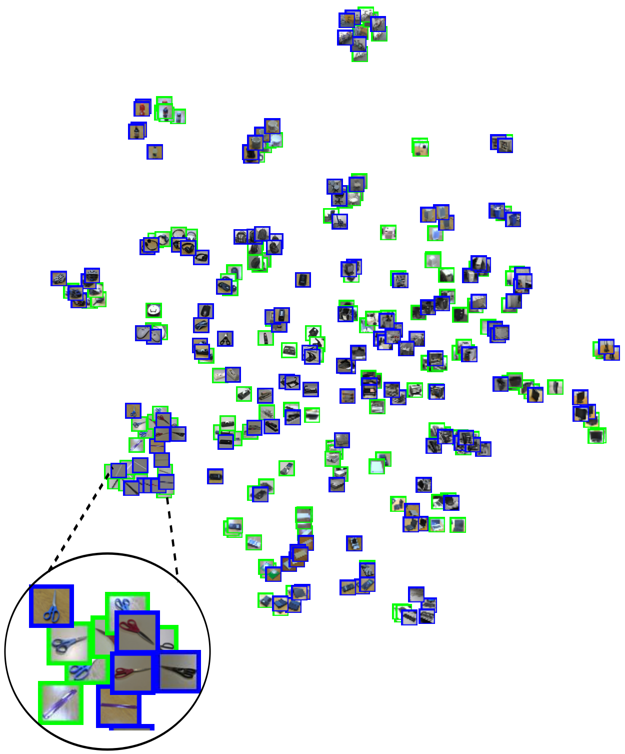
...but not completely.

backpacks are separated

Solving domain shift?

CNNs already remove a lot of the domain shift in the Office dataset [1]

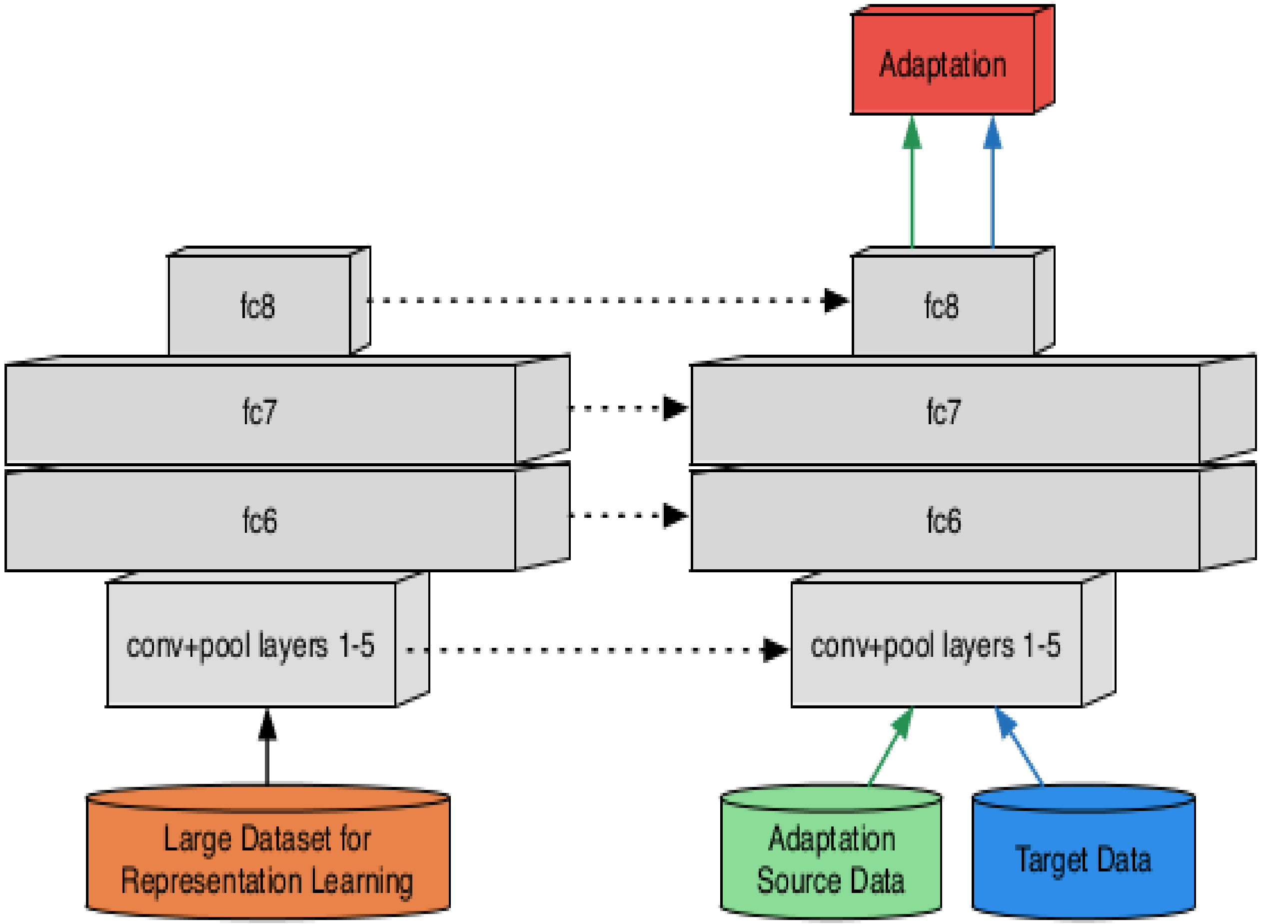
- But, can we get even more juice out of our network by tweaking it for the target domain?
- And, what if we just use a *really big* source domain?



Adaptation of Supervised Deep ConvNets with Limited Training Data

- Fully training a CNN takes a long time
- It would be nice to be able to reuse a trained model on a different task
- Typical solution: fine-tune!
  - But this only works when there is a lot of training data
- Hoffman et al. [arXiv 2013 / ICLR Workshop 2014]
  - *We provide the first analysis of standard domain adaptation techniques with CNNs to handle the case where target domain training data is severely limited*
  - *We also provide the first investigation of domain adaptation using a large-scale source domain (ImageNet!)*

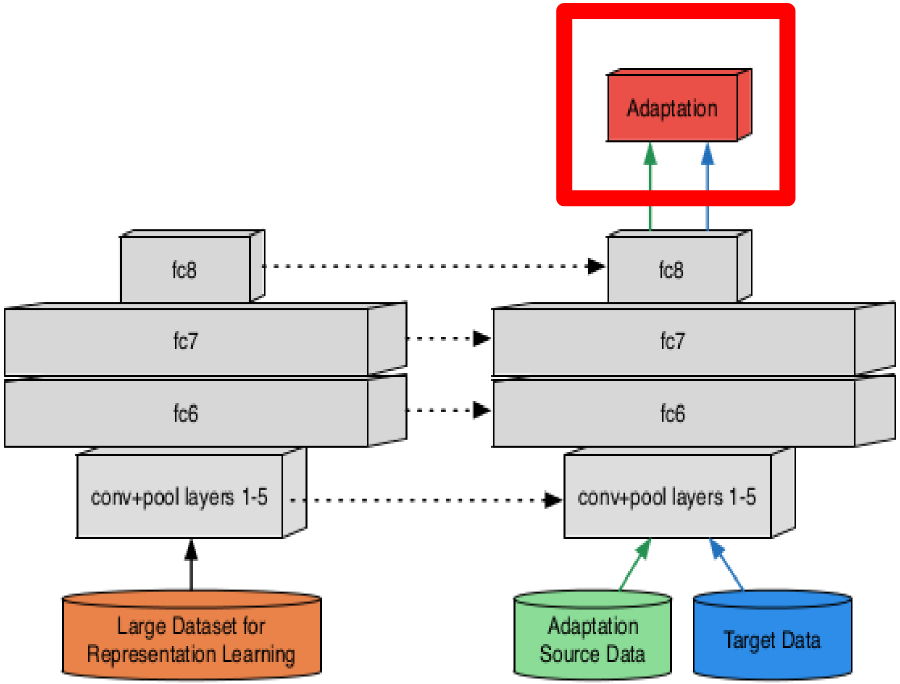
Network Adaptation Framework



[Hoffman et al. arXiv 2013 / ICLR Workshop 2014]

Adaptation Methods

- Three proposed methods of adaptation:
  - Deep and Frustratingly Easy (DFE)
  - Deep Late Fusion (DLF)
  - Deep Subspace Alignment (DSA)
- Each has its own advantages and disadvantages…



[Hoffman et al. arXiv 2013 / ICLR Workshop 2014]

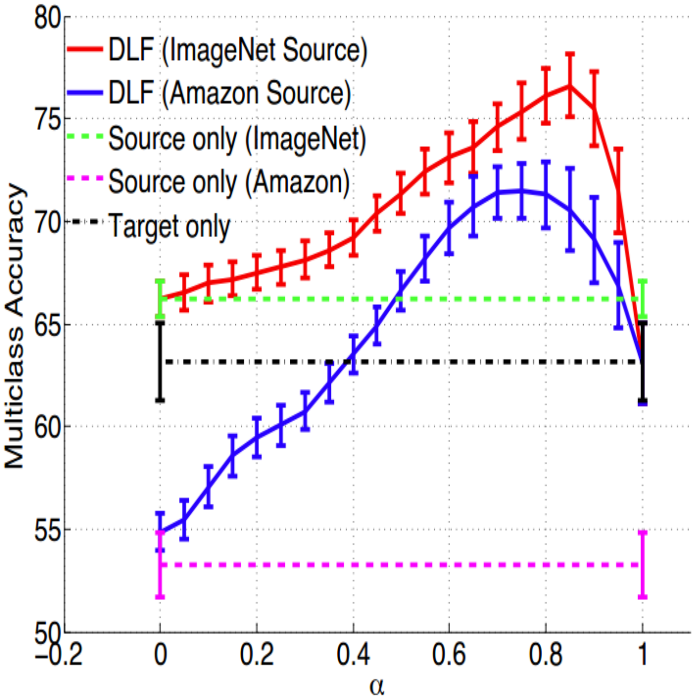
Deep and Frustratingly Easy (DFE)

- Supervised adaptation technique introduced by Daumé III
- Augment feature space by introducing three components: source-specific, target-specific, and common
  - e.g. Source data has the original feature vector replicated in the source-specific and common components and zeros in the target-specific component
- Then train a classifier on the augmented features
- Usually yields good performance
- …But requires all source and target data at train time

[2] H. Daumé III. Frustratingly easy domain adaptation. In ACL, 2007.

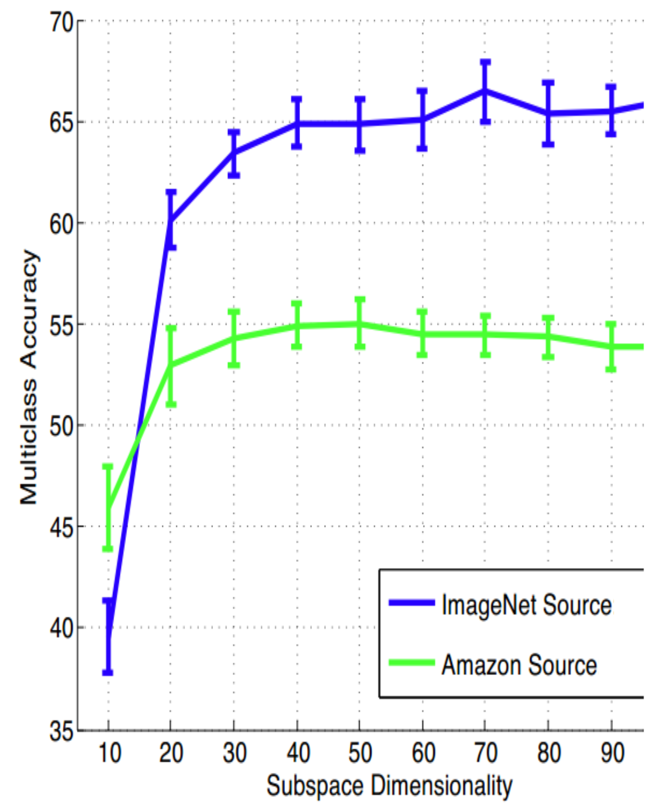
Deep Late Fusion (DLF)

- Train classifiers for source and target domains separately
- Final classifier score is a linear interpolation of the two separate source/target classifier scores
- Source classifier can be trained in advance and saved
- …But the linear interpolation coefficient must be set
  - (0.8 works pretty well in practice)



Deep Subspace Alignment (DSA)

- Unsupervised adaptation
- Compute source and target subspaces, and find an alignment between them
  - Train a classifier on the source data
  - At test time, project target data into source subspace and use source classifier
- Requires no labels, source subspace can be computed in advance



[3] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In Proc. ICCV, 2013.

Supervised setting results

	$A \to D$	$A \to W$	$D \to A$	$D \to W$	$W \to A$	$W \to D$	Average
GFK(PLS,PCA) [13]	-	46.4 ± 0.5	-	61.3 ± 0.4	-	66.3 ± 0.4	53.0
SA [11]	-	45.0	-	64.8	-	69.9	59.9
DLID [6]	-	51.9	-	78.2	-	**89.9**	73.3
DA-NBNN [23]	-	52.8 ± 3.7	-	76.6 ± 1.7	-	76.2 ± 2.5	68.5
Ours - source only	50.2 ± 0.6	45.6 ± 0.5	45.3 ± 0.3	$\mathbf{86.5 \pm 0.3}$	44.2 ± 0.3	88.0 ± 0.4	73.4
Ours - target only	77.5 ± 0.5	76.0 ± 0.5	55.8 ± 0.5	76.0 ± 0.5	55.8 ± 0.5	77.5 ± 0.5	76.5
Ours - DFE	$\mathbf{78.5 \pm 0.5}$	$\mathbf{77.4 \pm 0.4}$	$\mathbf{58.4 \pm 0.5}$	85.2 ± 0.3	$\mathbf{59.1 \pm 0.4}$	87.0 ± 0.4	**83.2**
Ours - DLF	69.3 ± 0.8	66.8 ± 0.7	56.7 ± 0.4	86.4 ± 0.4	56.3 ± 0.6	87.1 ± 0.6	80.1

Source: 20 images/category for Webcam and DSLR
8 images/category for Amazon
Target: 3 images/category

[Hoffman et al. arXiv 2013 / ICLR Workshop 2014]

Unsupervised setting results

	$A \rightarrow D$	$A \rightarrow W$	$D \rightarrow A$	$D \rightarrow W$	$W \rightarrow A$	$W \rightarrow D$	Average
GFK(PLS,PCA) [13]	-	15.0 ± 0.4	-	44.6 ± 0.3	-	49.7 ± 0.5	36.4
SA [11]	-	15.3	-	50.1	-	56.9	40.8
DA-NBNN [23]	-	23.3 ± 2.7	-	67.2 ± 1.9	-	67.4 ± 3.0	52.6
DLID [6]	-	26.1	-	68.9	-	84.9	60.0
Ours - source only	$\mathbf{50.2 \pm 0.6}$	45.6 ± 0.5	45.3 ± 0.3	$\mathbf{86.5 \pm 0.3}$	$\mathbf{44.2 \pm 0.3}$	$\mathbf{88.0 \pm 0.4}$	**73.4**
Ours - DSA	50.1 ± 0.4	$\mathbf{46.6 \pm 0.7}$	$\mathbf{45.5 \pm 0.4}$	86.2 ± 0.3	43.0 ± 0.4	86.7 ± 0.5	73.2

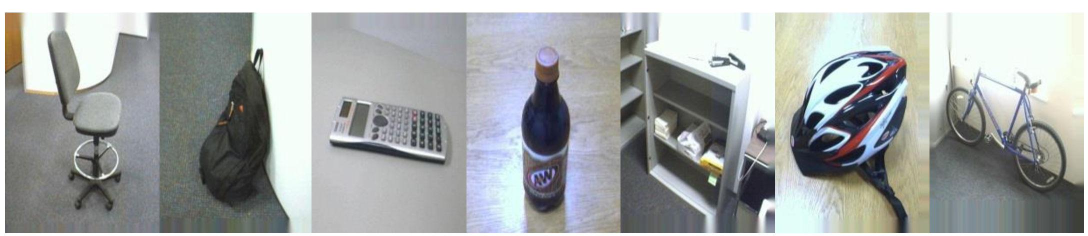
Source: 20 images/category for Webcam and DSLR
8 images/category for Amazon

[Hoffman et al. arXiv 2013 / ICLR Workshop 2014]

Large-scale Source Domains

- What happens if we try using a much bigger source dataset instead?
- Intuitively: a sufficiently large dataset should contain images like the target domain too!

IMAGENET



Webcam/DSLR

[Hoffman et al. arXiv 2013 / ICLR Workshop 2014]

Using ImageNet as Source

Adaptation Method	Source Domain Imagenet	Amazon
source only	55.5 ± 0.2	45.2 ± 1.4
DSA	55.4 ± 0.4	46.1 ± 2.0

(a) Unsupervised Adaptation

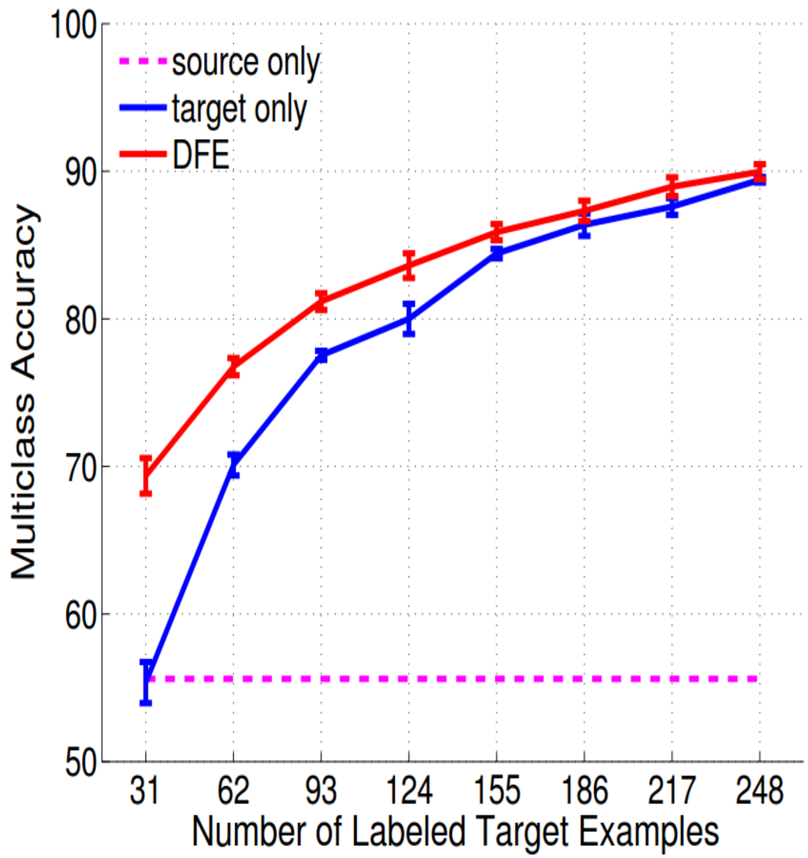
Adaptation Method	Source Domain ImageNet	Amazon
source only	55.5 ± 0.2	45.2 ± 1.4
target only	55.5 ± 2.7	55.5 ± 2.7
DLF	62.8 ± 1.2	56.8 ± 2.3
DFE	$\mathbf{69.9 \pm 1.5}$	64.5 ± 1.3

(b) Supervised Adaptation (One-Shot Adaptation)

Evaluation on Webcam domain

[Hoffman et al. arXiv 2013 / ICLR Workshop 2014]

When to Adapt?



ImageNet to Webcam

[Hoffman et al. arXiv 2013 / ICLR Workshop 2014]

Outline

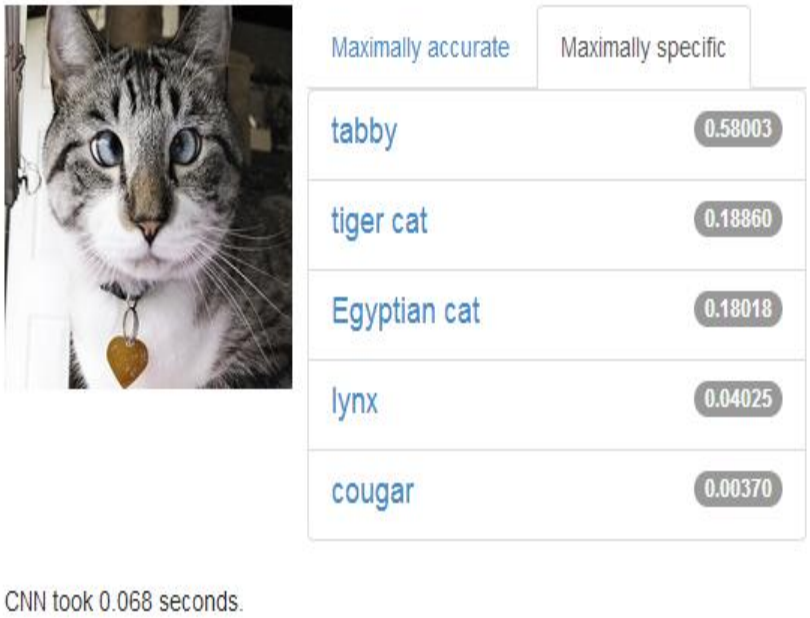
- **What does deep learning offer domain adaptation and vice-versa?**
- A domain-adaptation perspective on large-scale ImageNet detection
- Beyond binary domain adaptation

Outline

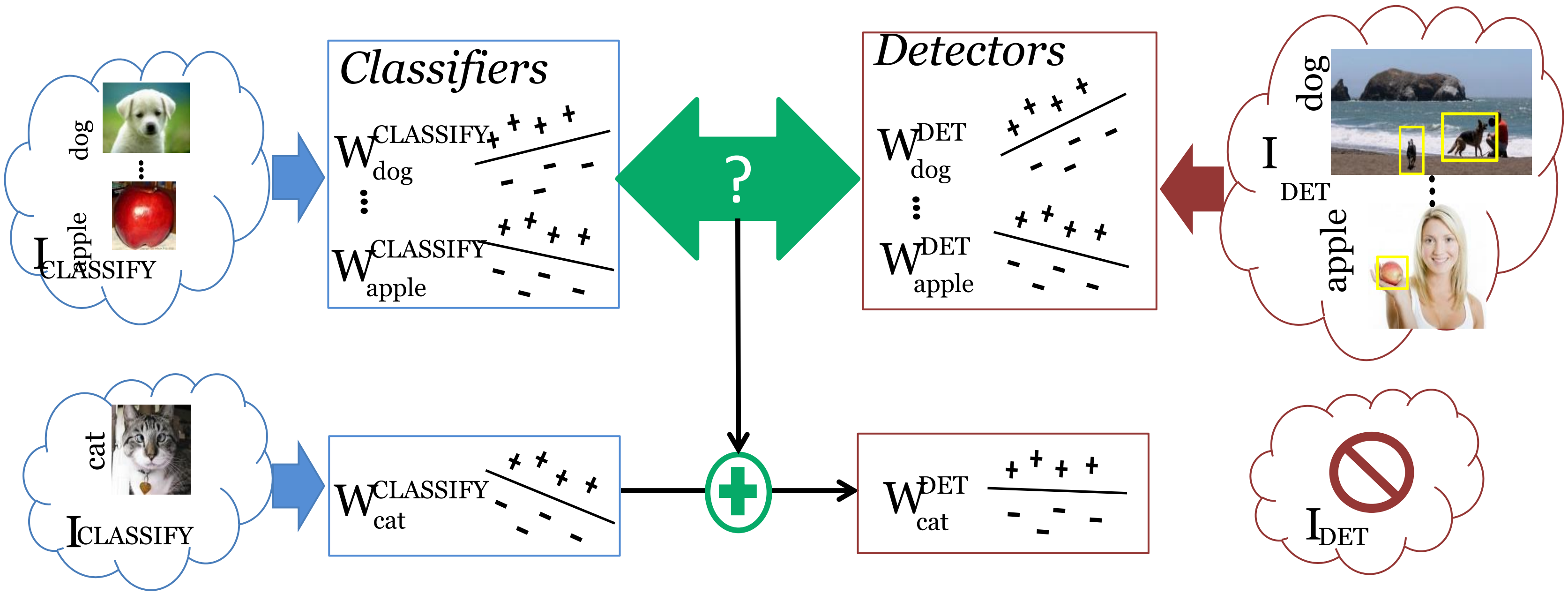
- What does deep learning offer domain adaptation and vice-versa?
- **A domain-adaptation perspective on large-scale ImageNet detection**
- Beyond binary domain adaptation

Large-scale detection

- ImageNet provides huge amounts of classification data, so classification performance is very good even with a large number of categories
- But there is comparatively little detection data, since bounding boxes are hard to come by.
  - How to learn detectors for thousands of categories?



Idea: adapt classifiers for detection



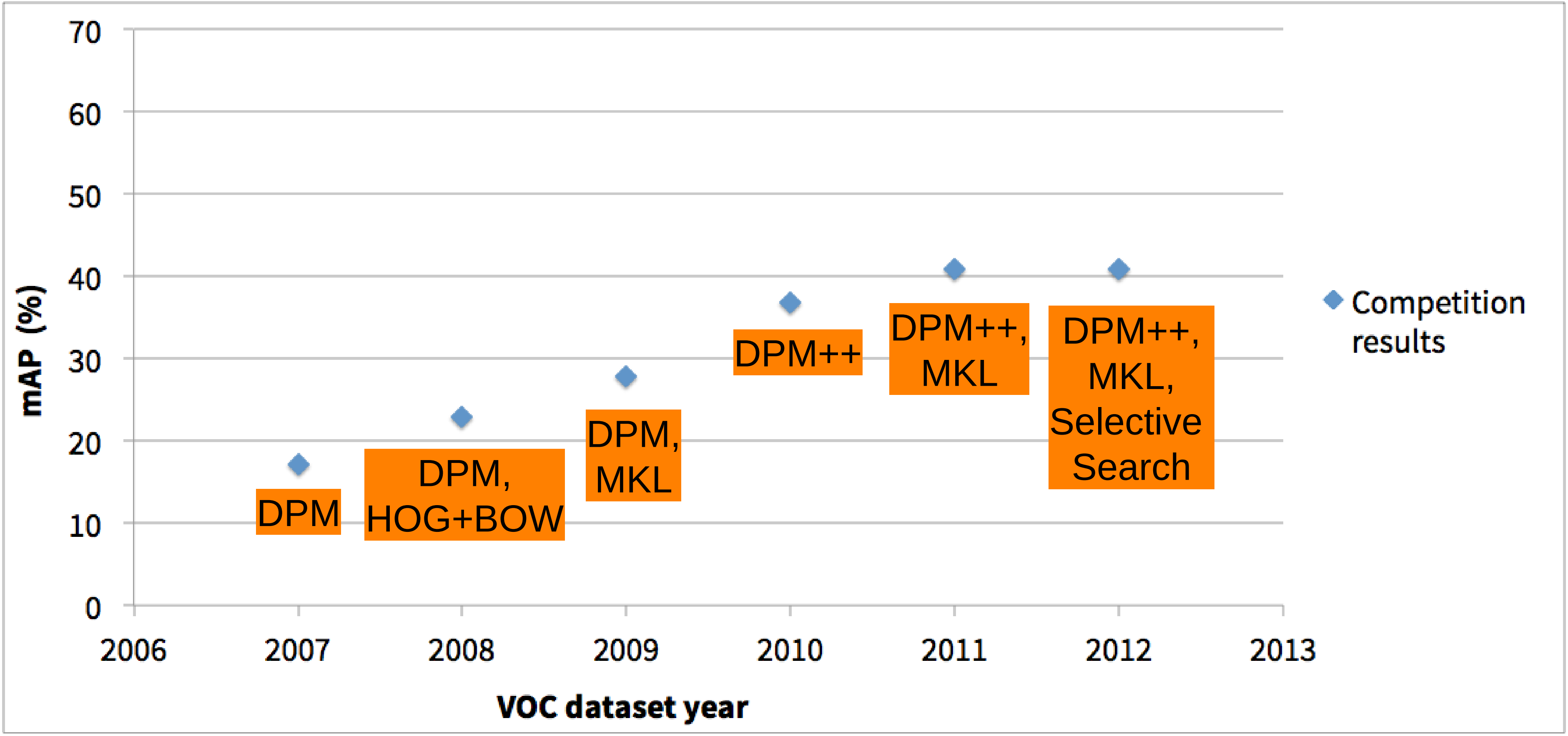
[Hoffman et al. arXiv 2014 / NIPS 2014]

PASCAL VOC Challenge

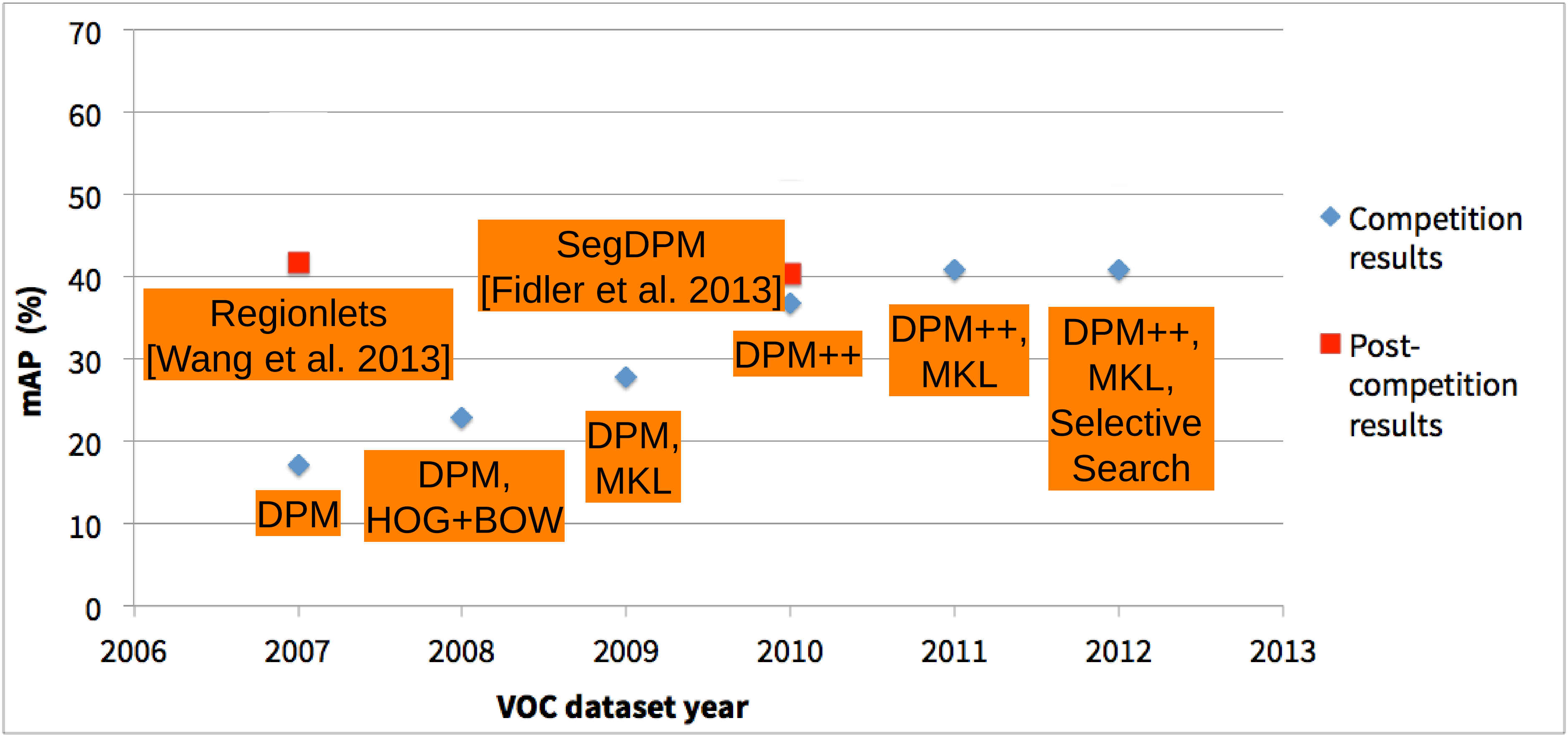
Dataset: 22k images, 50k objects, 20 classes

Detect: people, horses, sofas, bicycles, pottedplants

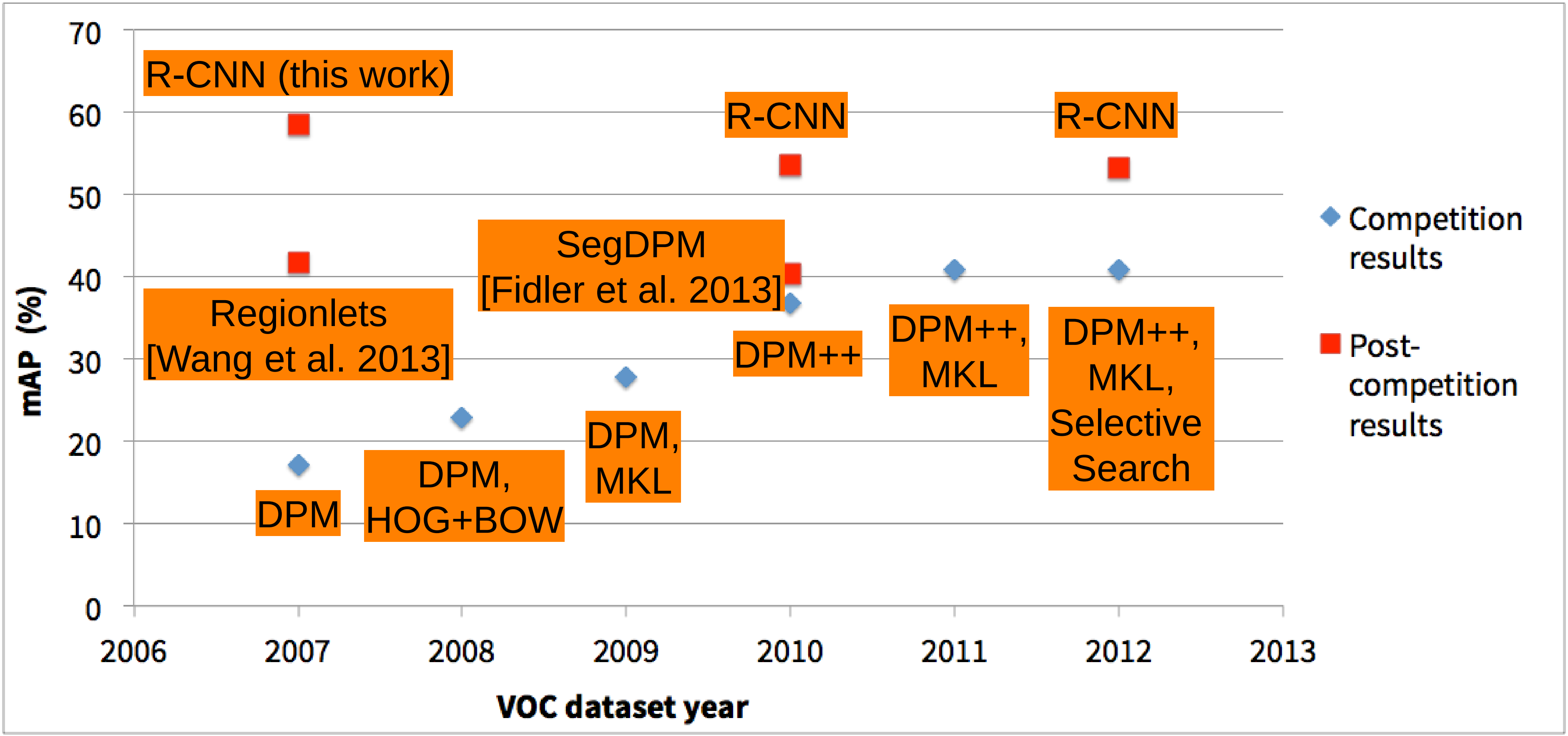
Progress on PASCAL VOC



Progress on PASCAL VOC



Progress on PASCAL VOC



ImageNet LSVR Challenge

– 1000 classes
  (vs. 20)

– 1.2 million training images
  (vs. 10k)

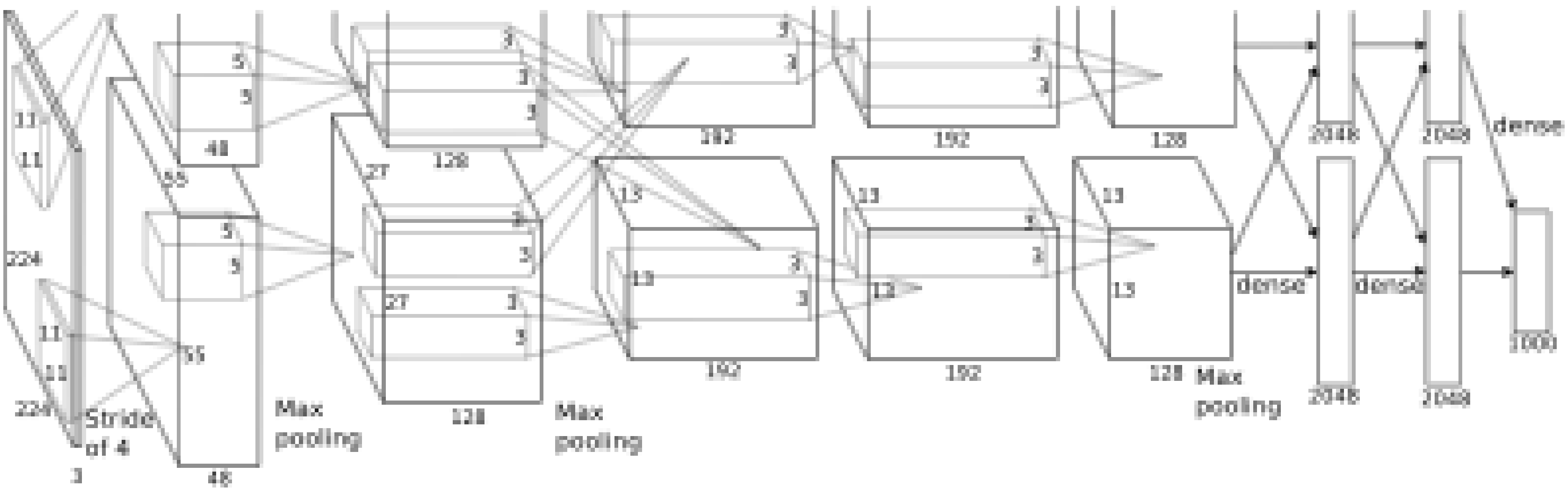
– Image classification
  (not detection)

bus anywhere?

[Deng et al. CVPR'09]

Multi-layer feature learning

"SuperVision" Convolutional Neural Network (CNN)



input ← 5 convolutional layers → fully connected

ImageNet Classification with Deep Convolutional Neural Networks.
Krizhevsky, Sutskever, Hinton. NIPS 2012.

cf. LeCun et al. Neural Comp. '89 & Proc. of the IEEE '98

Impressive ImageNet results!

1000-way image classification

	Top-5 error
Fisher Vectors (ISI)	26.2%
5 SuperVision CNNs	16.4%
7 SuperVision CNNs	15.3%

now: 12%

metric: classification error rate (lower is better)

But... does it generalize to other datasets and tasks?

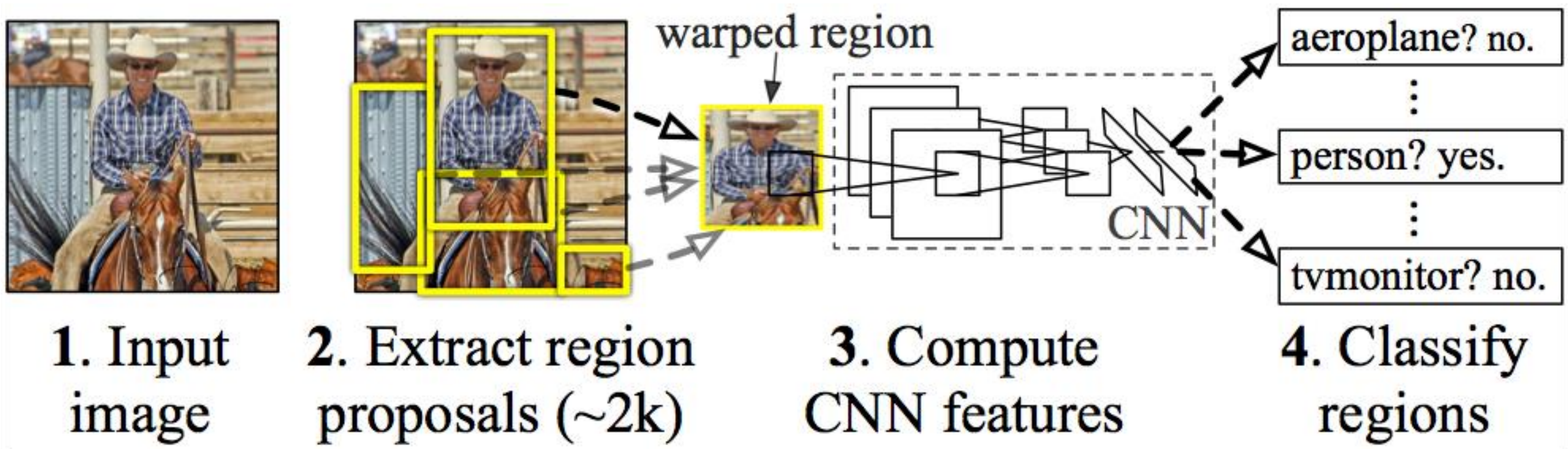
Spirited debate at ECCV 2012

Objective

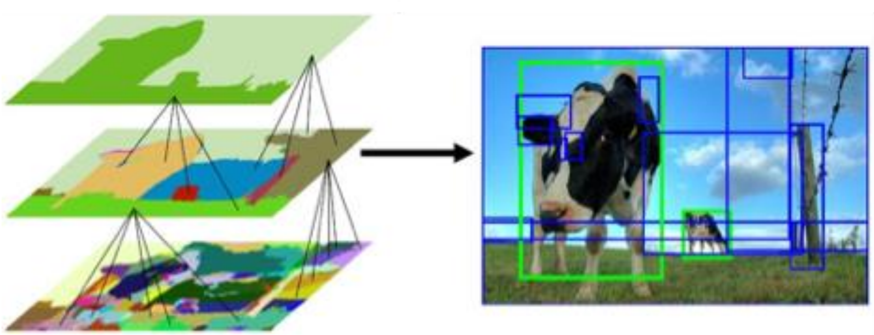
Can the SuperVision CNN detect objects?

Proposed system

R-CNN: “Regions with CNN features”



[Girshick, Donahue, Darrell, Malik
to appear in CVPR’14]



“selective search” [van de Sande et al. 2011]

R-CNN results on PASCAL

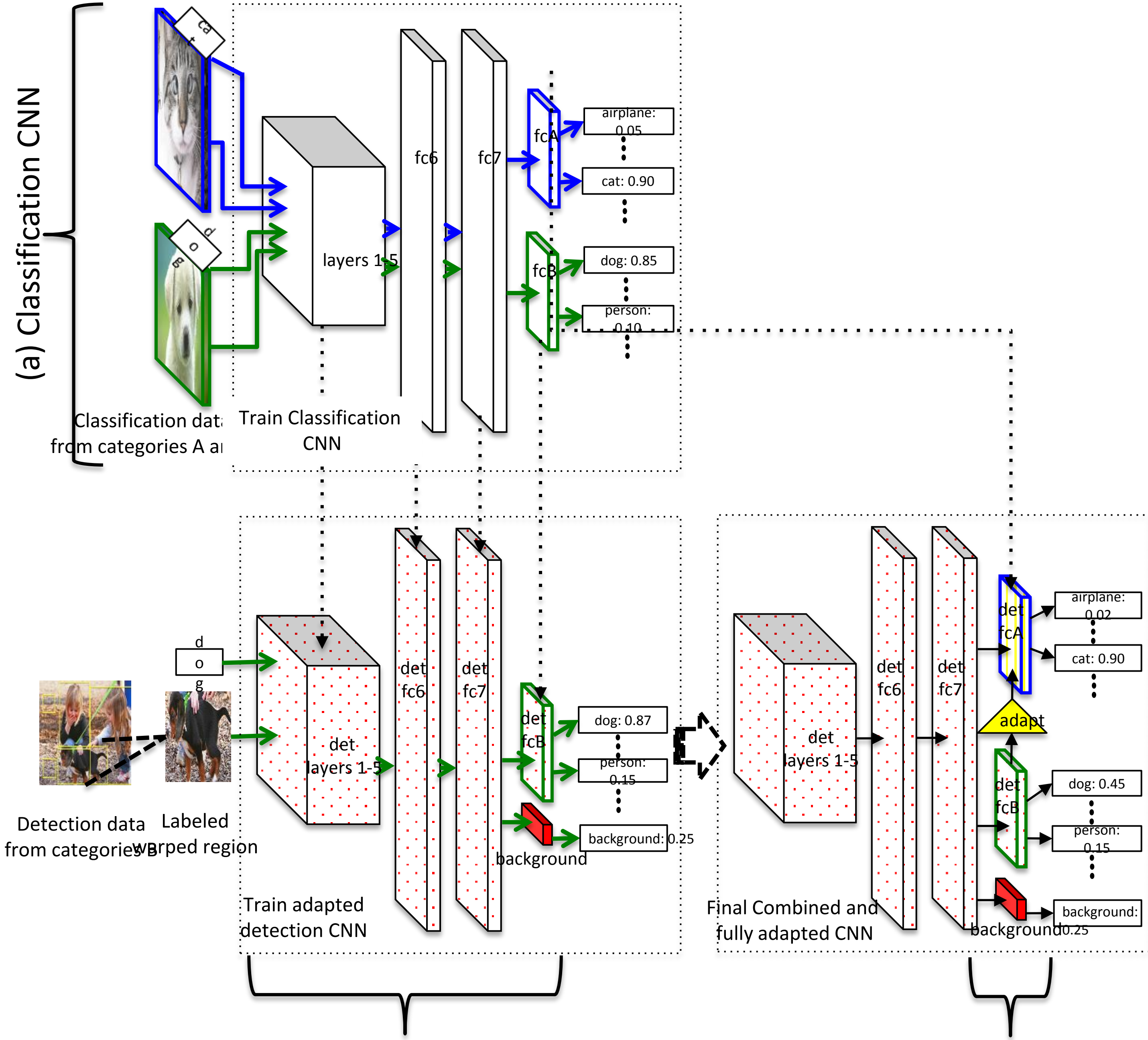
	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2012)		35.1%
Regionlets (Wang et al. 2013)	41.7%	39.7%

metric: mean average precision (higher is better)

R-CNN results on PASCAL

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2012)		35.1%
Regionlets (Wang et al. 2013)	41.7%	39.7%
R-CNN	54.2%	50.2%
R-CNN + bbox regression	58.5%	53.7%

metric: mean average precision (higher is better)



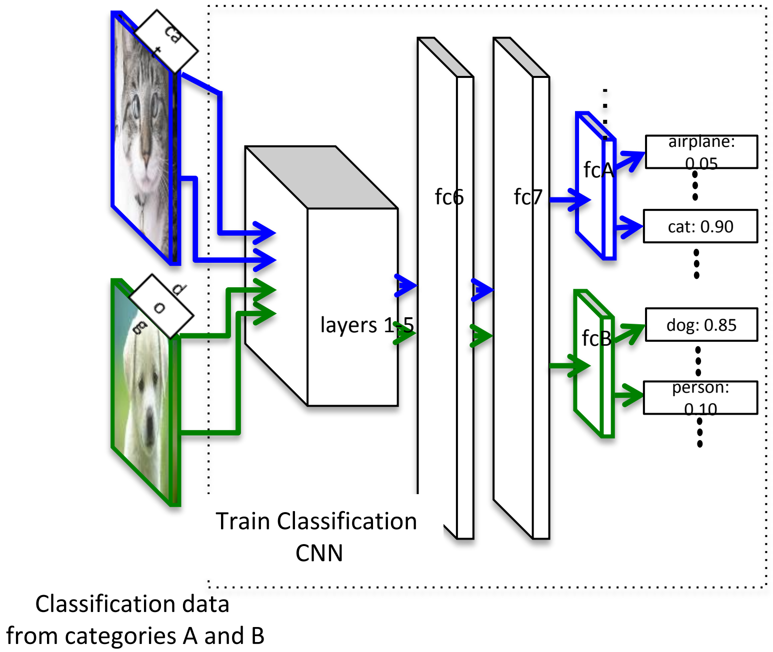
(b) Hidden Layer Adaptation

(c) Output Layer Adaptation

[Hoffman et al. arXiv 2014 / NIPS 2014]

Classification CNN

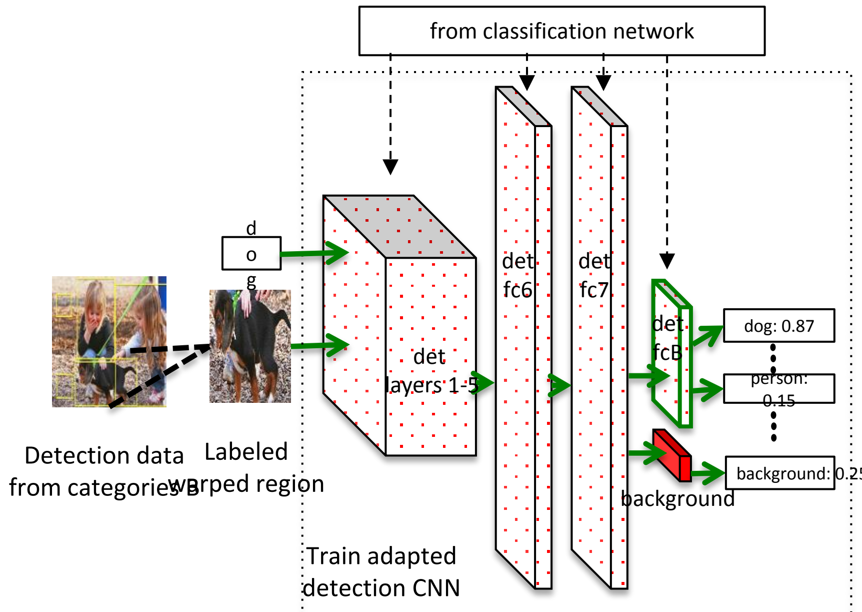
- Train a classification network using your favorite architecture
- We begin with Krizhevsky et al.'s network and train it on the 1000-way ImageNet classification task
- We then replace the final layer with a new multinomial logistic regression layer to recognize the categories we are interested in



[Hoffman et al. arXiv 2014 / NIPS 2014]

Hidden Layer Adaptation

- For the classes with detection data, we now train detectors
- Begin with weights from classification network
- Add a background class
  - Gathered by sampling selective search windows with low overlap with ground truth bounding boxes
- Fine-tune network using detection data + background patches

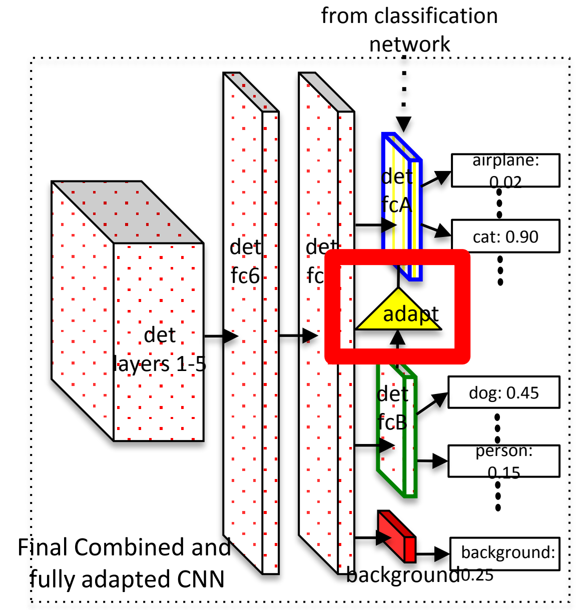


[Hoffman et al. arXiv 2014 / NIPS 2014]

Output Layer Adaptation

- Directly transferring the output layer for the classes without detection data is a bad idea – preceding layers have changed!
- Simple idea: compute average classification-to-detection change for each class in the set of classes with detection data *B*:

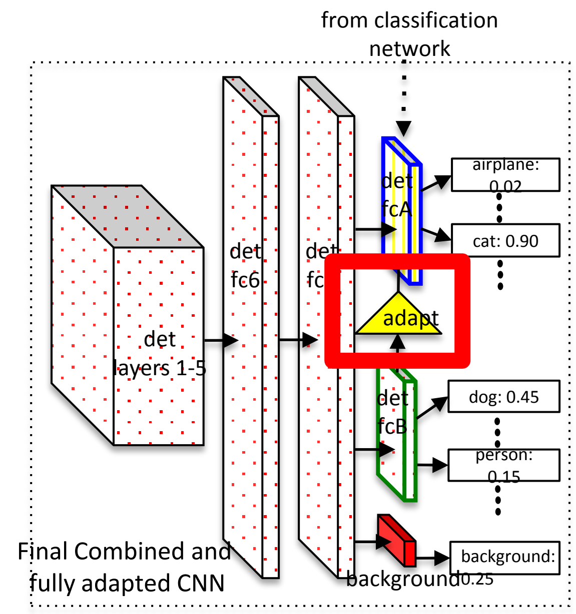
$$\Delta W_{avg} = \frac{1}{|B|}\sum_{i\in B} W_i^d - W_i^c.$$



[Hoffman et al. arXiv 2014 / NIPS 2014]

Output Layer Adaptation

- Intuitively: some classes may transform differently than others
- Use the "closest" classes to determine the transformation
  - "Close" means small Euclidean distance between the L2 normalized FC7 layer parameters
- Letting $N_B(j,k)$ denote the k-th nearest neighbor
  to class j, we have:



$$\forall j \in A : W_j^d = W_j^c + \frac{1}{k}\sum_k \left[W^d_{N_B(j,k)} - W^c_{N_B(j,k)} \cdot \right]$$

[Hoffman et al. arXiv 2014 / NIPS 2014]

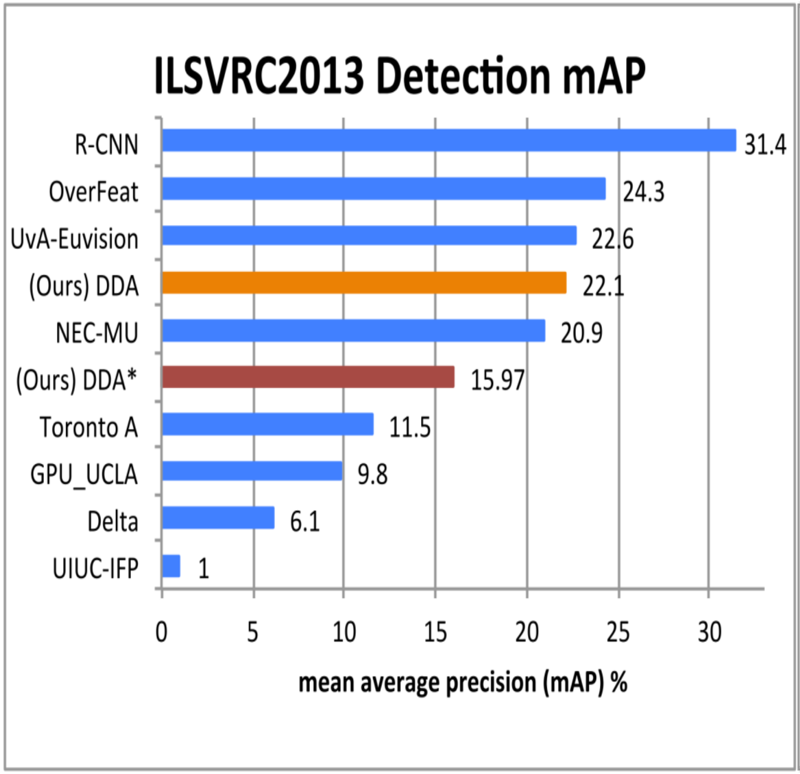
Results on ILSVRC2013

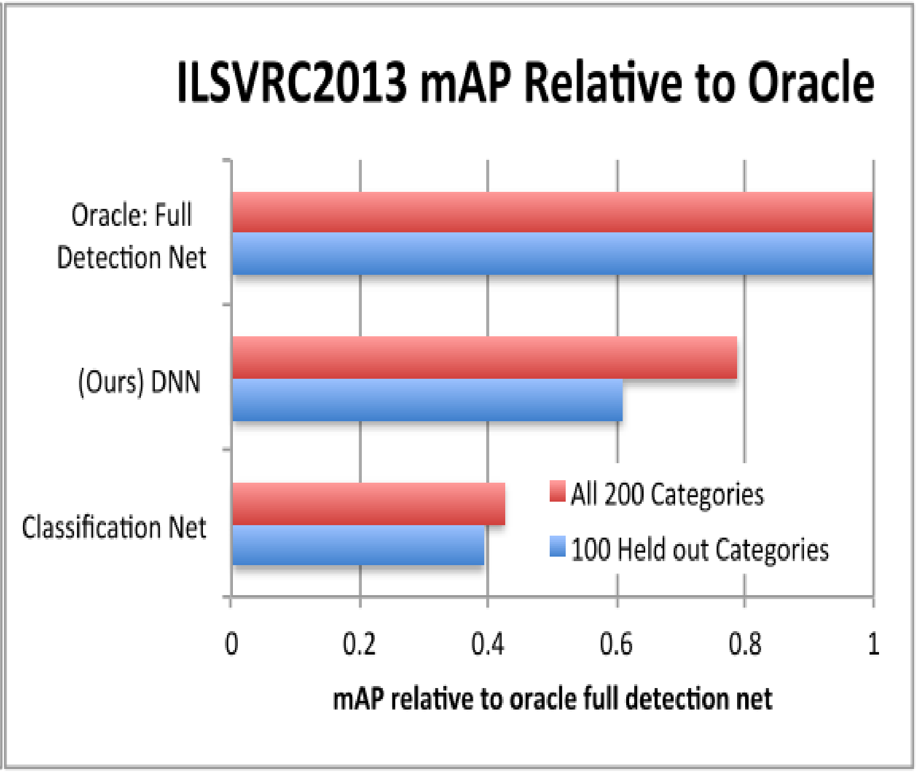
Detection Adaptation Layers	Output Layer Adaptation	mAP Trained 100 Categories	mAP Held-out 100 Categories	mAP All 200 Categories
No Adapt (Classification Network)		12.63	10.31	11.90
fc_{bgrnd}	-	14.93	12.22	13.60
fc_{bgrnd},fc_6	-	24.72	13.72	19.20
fc_{bgrnd},fc_7	-	23.41	14.57	19.00
fc_{bgrnd},fc_B	-	18.04	11.74	14.90
fc_{bgrnd},fc_6,fc_7	-	25.78	14.20	20.00
fc_{bgrnd},fc_6,fc_7,fc_B	-	26.33	14.42	20.40
fc_{bgrnd},layers1-7,fc_B	-	27.81	15.85	21.83
fc_{bgrnd},layers1-7,fc_B	Avg NN (k=5)	**28.12**	**15.97**	22.05
fc_{bgrnd},layers1-7,fc_B	Avg NN (k=100)	27.91	15.96	21.94
Oracle: Full Detection Network		29.72	26.25	28.00

Table 1: Ablation study for the pieces of DNN. We consider removing different pieces of our algorithm to determine which pieces are essential. We consider training with the first 100 (alphabetically) categories of the ILSVRC2013 detection validation set (on val1) and report mean average precision (mAP) over the 100 trained on and 100 held out categories (on val2). We find the best improvement is from fine-tuning all convolutional fully connected layers and using output layer adaptation.

[Hoffman et al. arXiv 2014 / NIPS 2014]

Results on ILSVRC2013



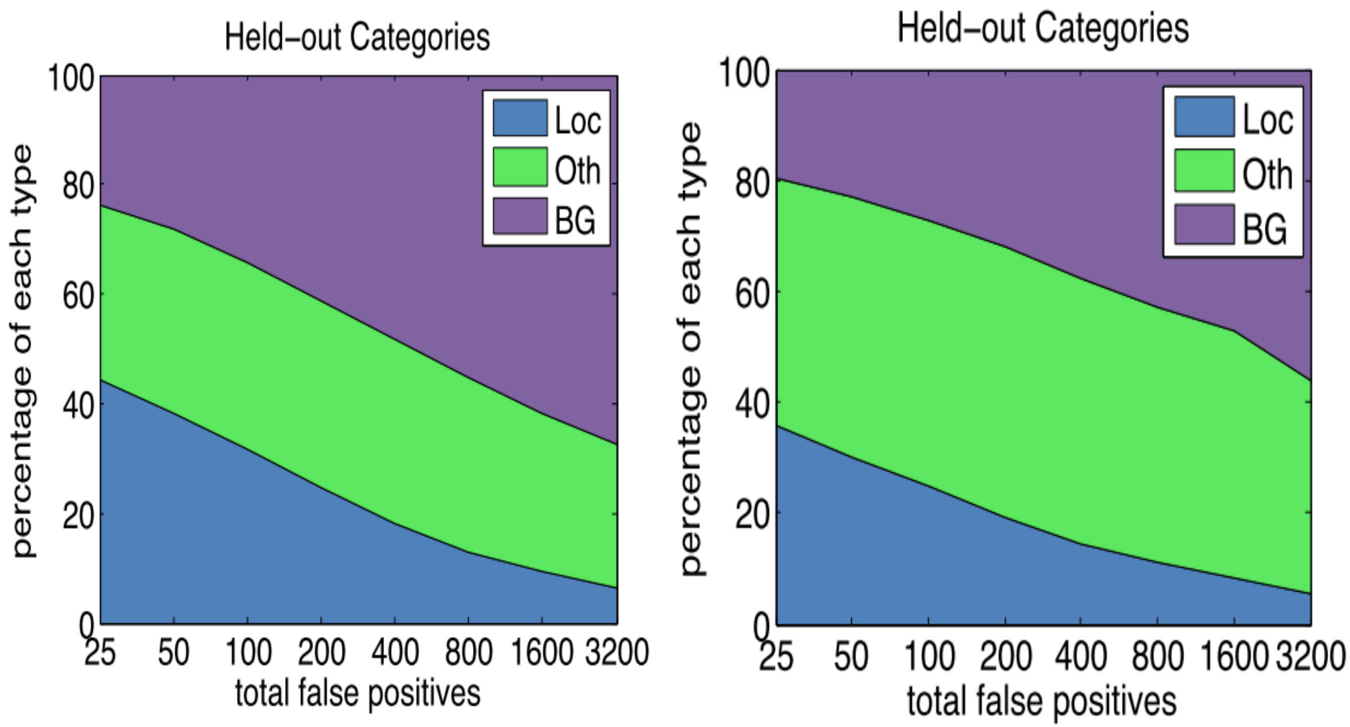


- Oracle is our method trained with all 200 category detection data

[Hoffman et al. arXiv 2014 / NIPS 2014]

Detailed analysis of error type

before adaptation after adaptation

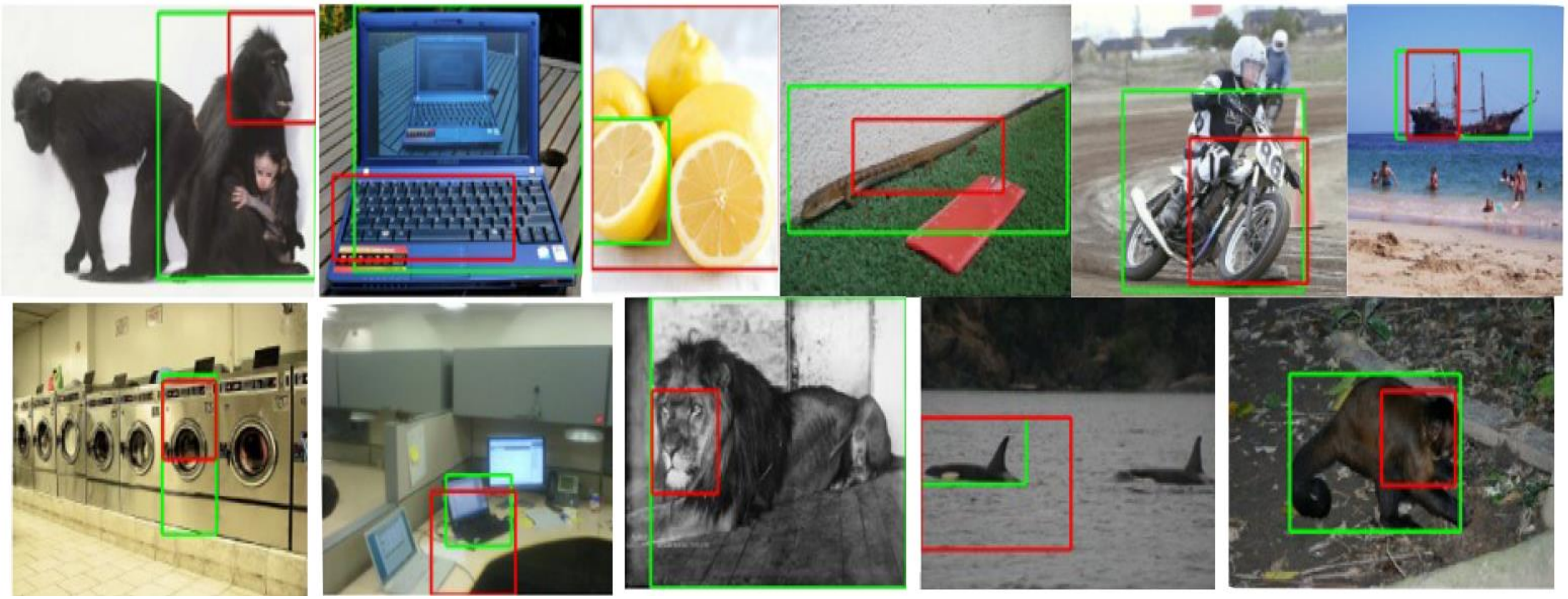


Loc: location error **Oth:** other error **BG:** confused with background

The adapted network makes fewer Loc and BG errors

[Hoffman et al. arXiv 2014 / NIPS 2014]

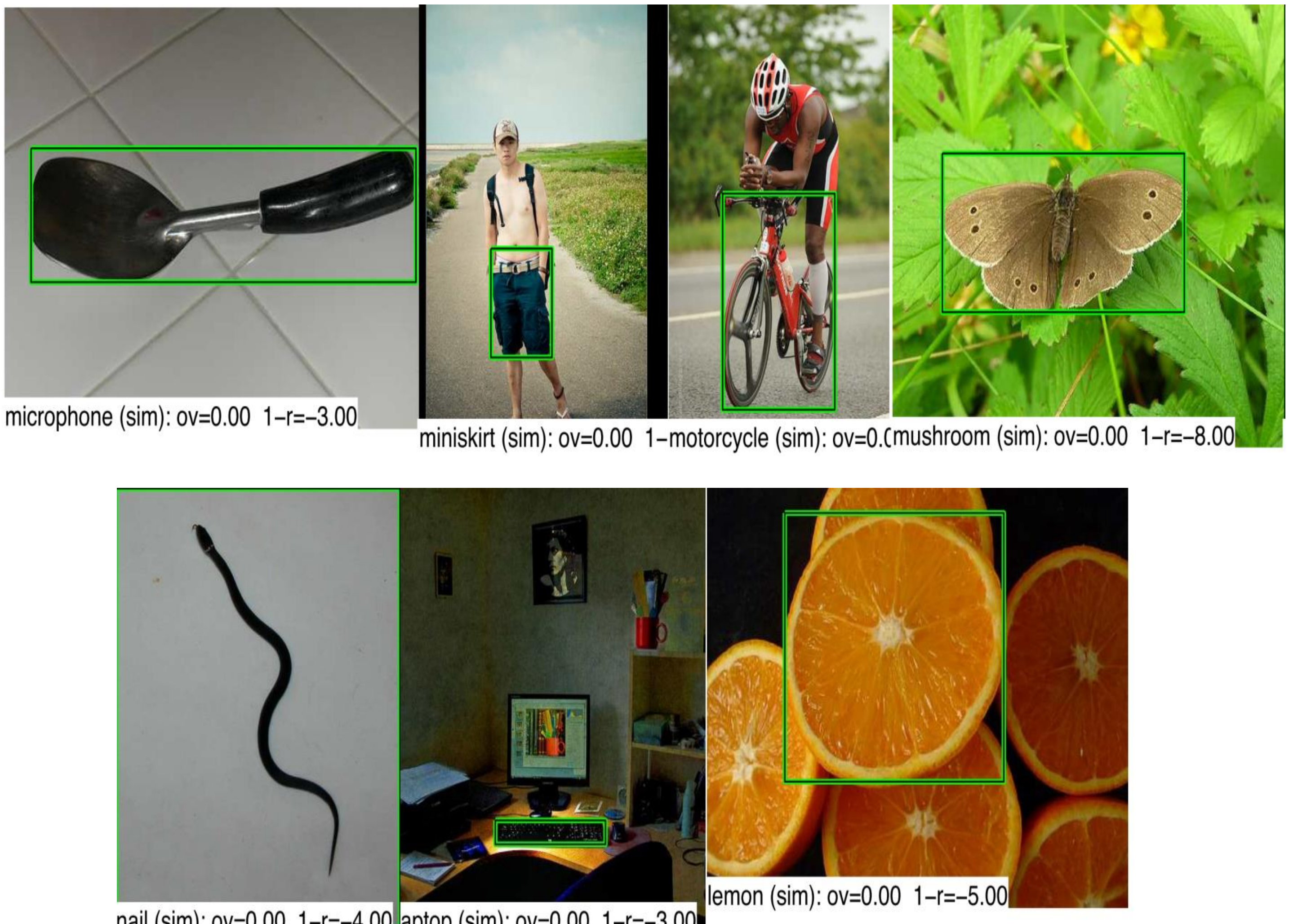
Better localization



Red: classification network before adaptation
Green: adapted

[Hoffman et al. arXiv 2014 / NIPS 2014]

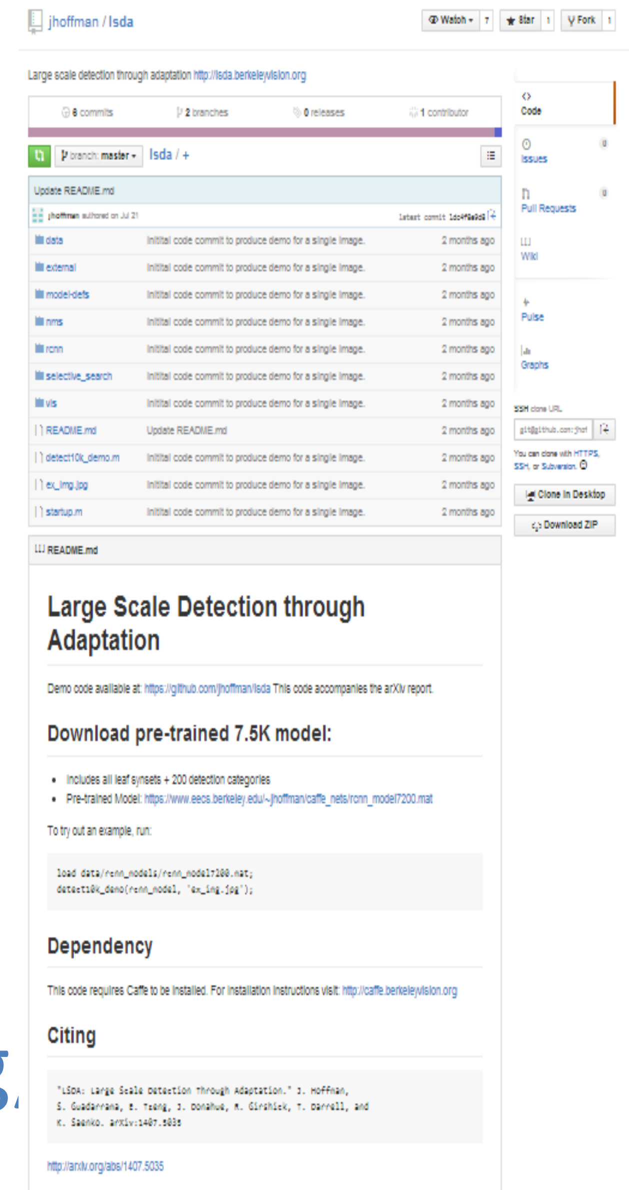
Failures


microphone (sim): ov=0.00 1−r=−3.00
miniskirt (sim): ov=0.00 1−
motorcycle (sim): ov=0.(
mushroom (sim): ov=0.00 1−r=−8.00
nail (sim): ov=0.00 1−r=−4.00
aptop (sim): ov=0.00 1−r=−3.00
lemon (sim): ov=0.00 1−r=−5.00

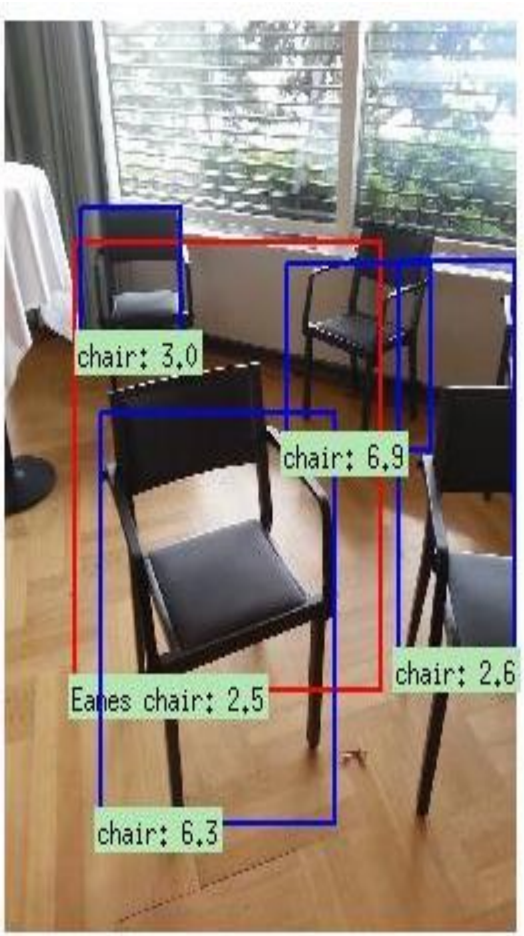
7K class detector!

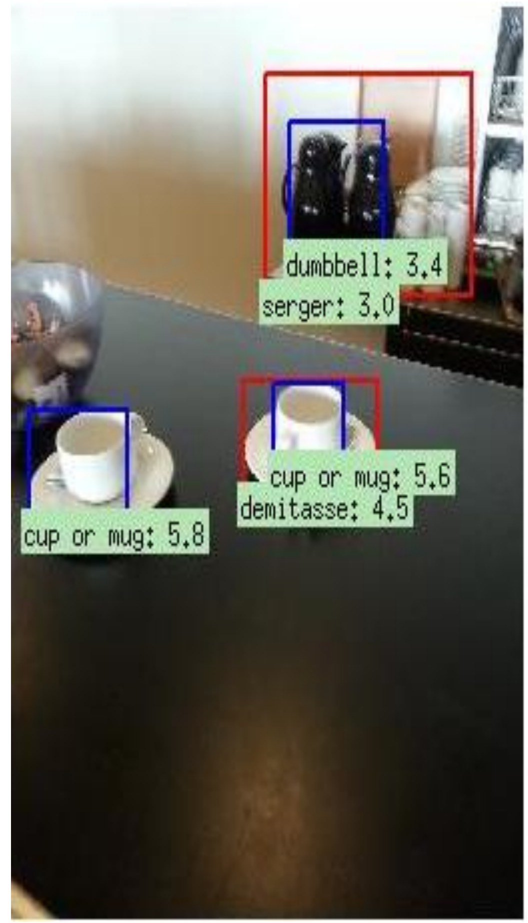
- Public release of a 7604 category detector trained using this method
  - 200 ILSVRC2013 classes trained with bounding box data
  - 7404 ImageNet leaf nodes trained with adaptation

http://lsda.berkeleyvision.org/

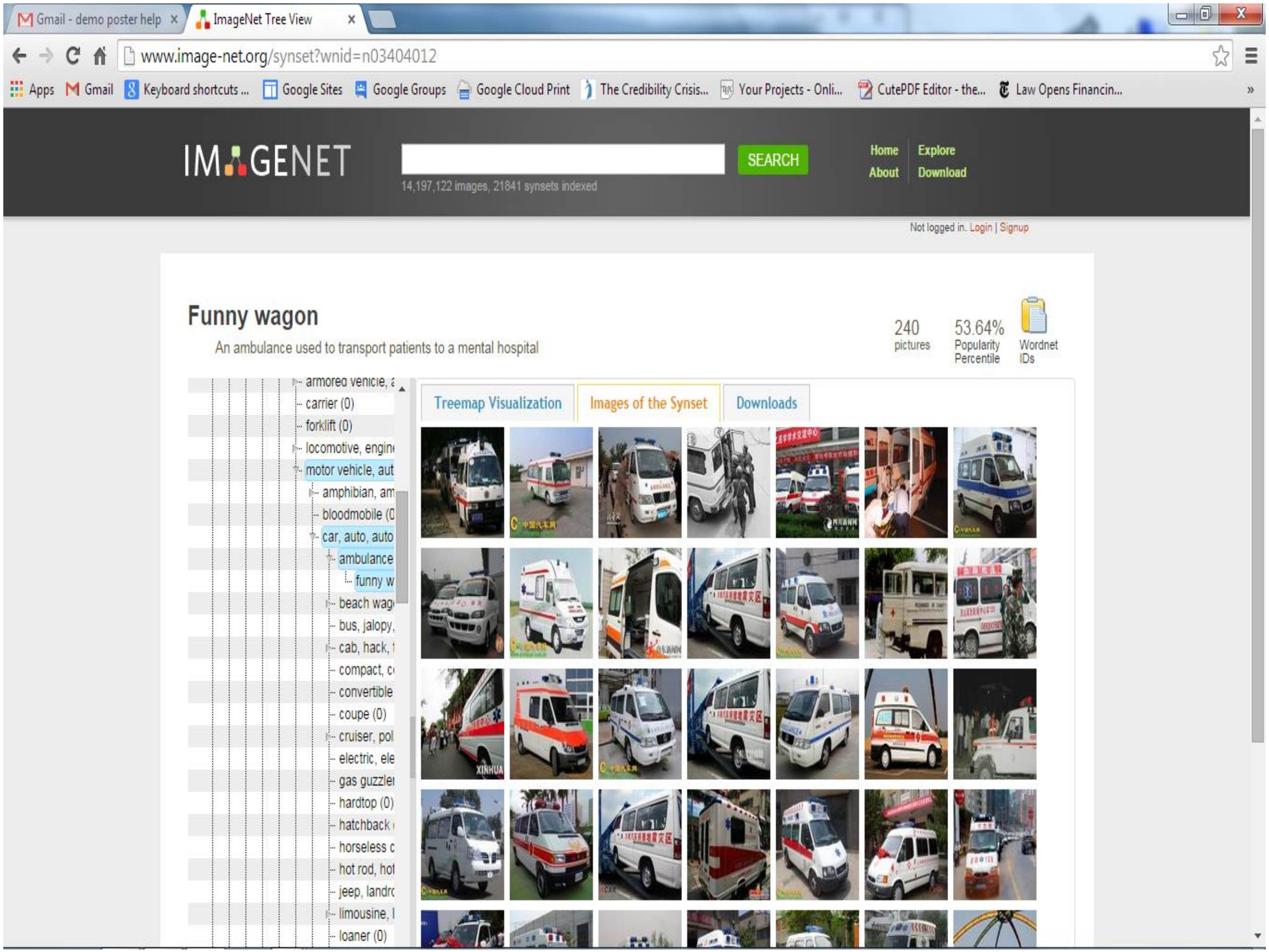


ECCV 2014 Demo...






car: 2.050
car: 1.267
funny wagon: 1.828
removable disk:
NOTRUF 144
electronic balance: 1.538


Gmail - demo poster help
ImageNet Tree View
www.image-net.org/synset?wnid=n03404012
Apps
Gmail
Keyboard shortcuts ...
Google Sites
Google Groups
Google Cloud Print
The Credibility Crisis...
Your Projects - Onli...
CutePDF Editor - the...
Law Opens Financin...
IMAGENET
SEARCH
14,197,122 images, 21841 synsets indexed
Home
Explore
About
Download
Not logged in. Login | Signup
Funny wagon
An ambulance used to transport patients to a mental hospital
240
pictures
53.64%
Popularity
Percentile
Wordnet
IDs
Treemap Visualization
Images of the Synset
Downloads
carrier (0)
forklift (0)
locomotive, engin
motor vehicle, aut
amphibian, am
bloodmobile (0
car, auto, auto
ambulance
funny w
beach wag
bus, jalopy,
cab, hack,
compact, c
convertible
coupe (0)
cruiser, pol
electric, ele
gas guzzle
hardtop (0)
hatchback
horseless c
hot rod, ho
jeep, landr
limousine, l
loaner (0)
XINHUA

Outline

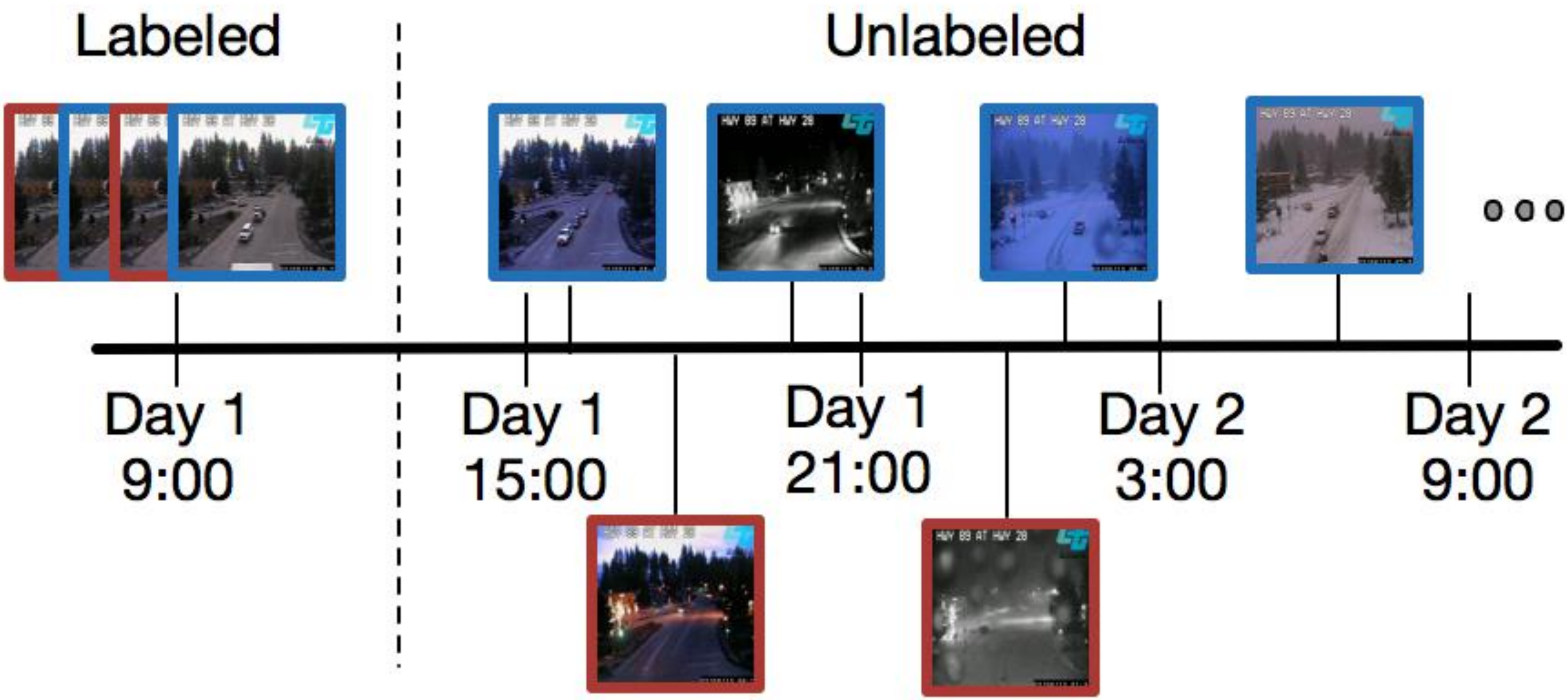
- What does deep learning offer domain adaptation and vice-versa?
- **A domain-adaptation perspective on large-scale ImageNet detection**
- Beyond binary domain adaptation

Outline

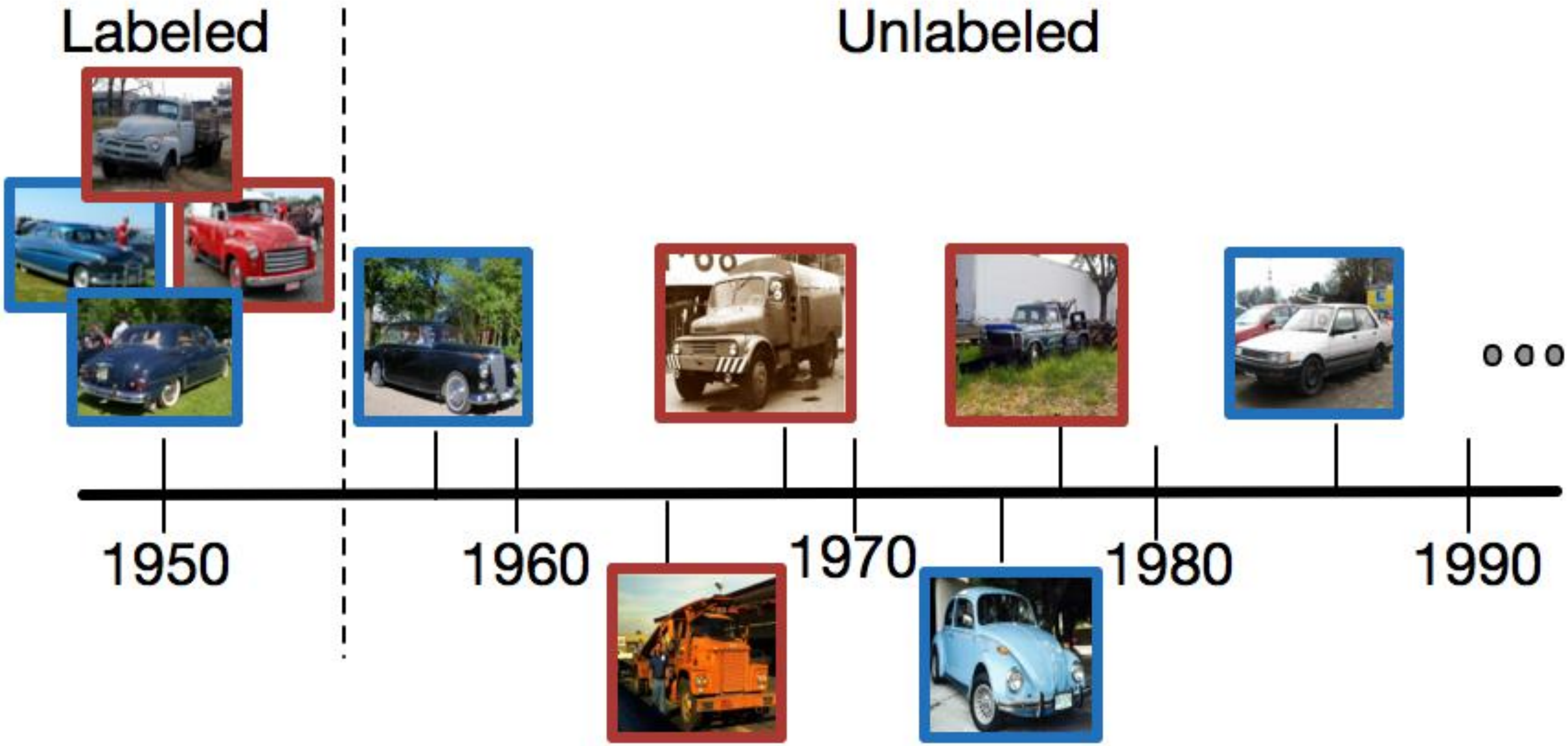
- What does deep learning offer domain adaptation and vice-versa?
- A domain-adaptation perspective on large-scale ImageNet detection
- **Beyond binary domain adaptation**

Beyond Binary Domains....

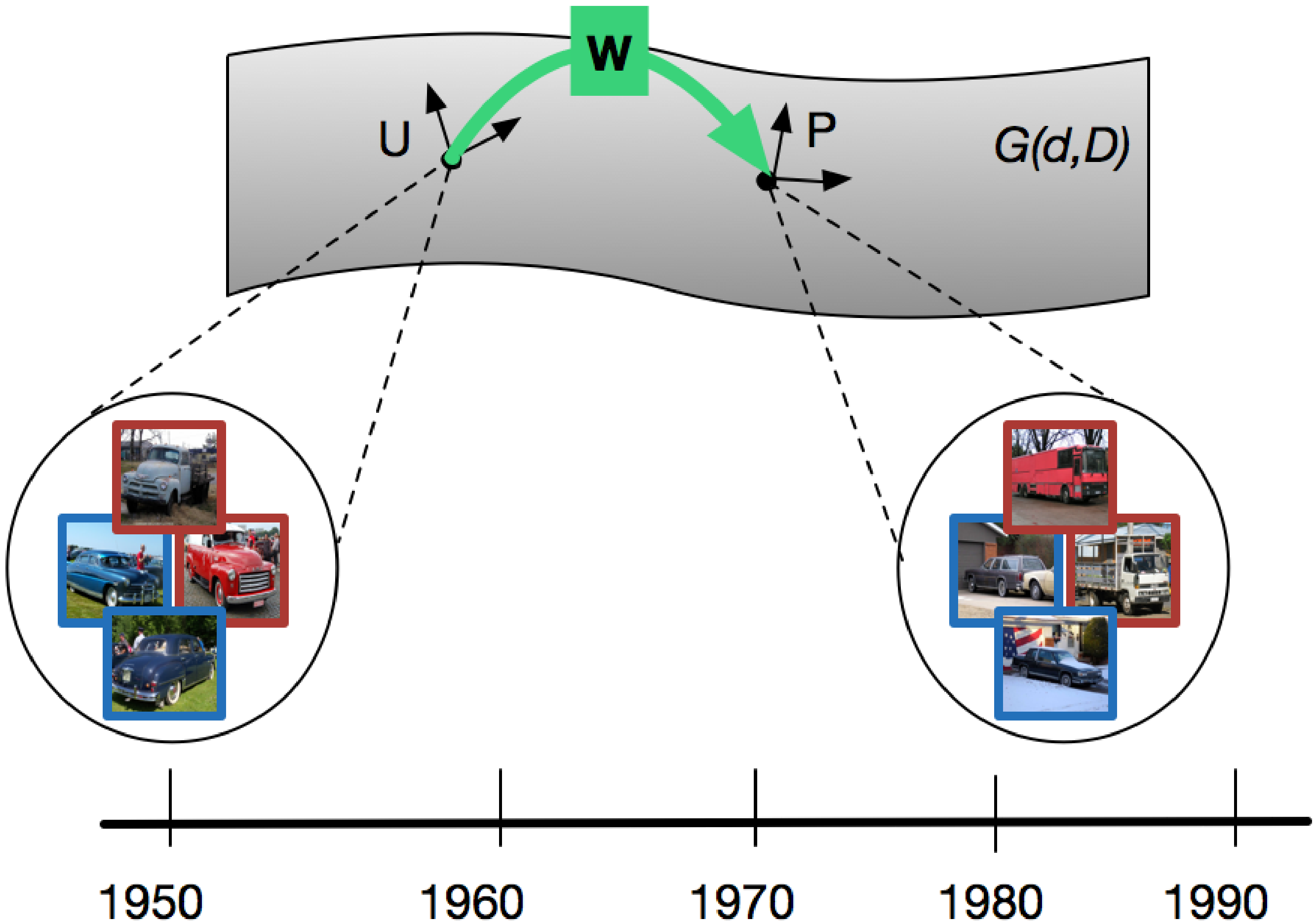
Traffic Intersection Data



Automobiles across decades

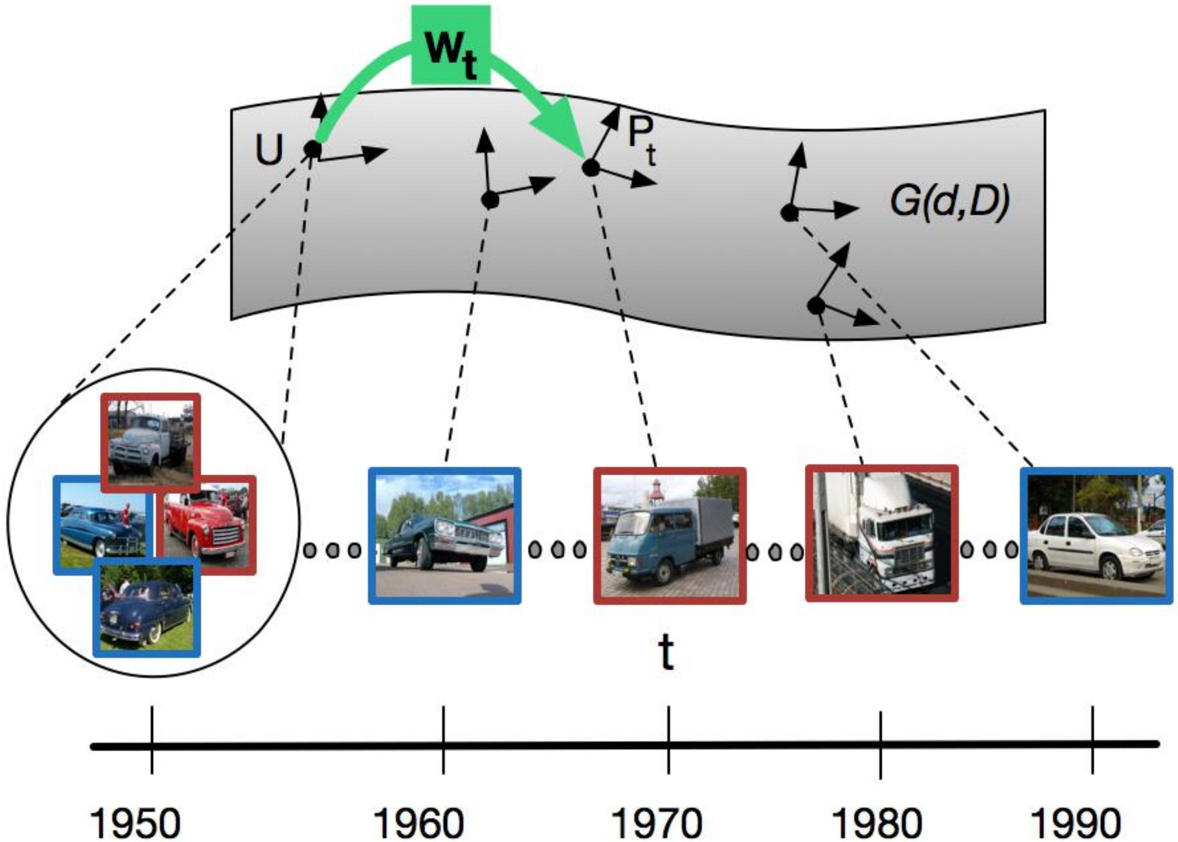


Unsupervised Adaptation



(Gopalan, ICCV 11), (Gong, CVPR 12),
(Fernando, ICCV 13)

Continuous Unsupervised Adaptation



$$\min_{P_t^T P_t = I, \tilde{A}_t, \tilde{B}_t} r(P_{t-1}, P_t) + R_{err}(z_t, P_t) + \psi(U\tilde{A}_t, P_t\tilde{B}_t)$$

where $r(\cdot)$ is a regularizer that encourages the new subspace learned at time t to be close to the previous subspace of time $t-1$.

Continuous Manifold Based Adaptation for Evolving Visual Domains
[CVPR 2014]

Judy Hoffman
UC Berkeley, EECS
jhoffman@eecs.berkeley.edu

Trevor Darrell
UC Berkeley, EECS
trevor@eecs.berkeley.edu

Kate Saenko
UMass Lowell, CS
saenko@cs.uml.edu

Adaptation	Classifier	GIST[26]
-	KNN	71.24±5.7
-	SVM	80.40±0.6
CMA+GFK	KNN	77.21± 3.8
CMA+GFK	SVM	**84.17±1.5**
CMA+SA	KNN	78.61±3.3
CMA+SA	SVM	**84.32±1.4**

Table 2. Our method, CMA, continues to provide improvement for the scene classification task even when testing over the 5 days following the labeled training data. We show here average precision (%) for the 2400 test images following the 50 available labeled training images.

Adaptation Method	Classifier	GIST[26]	SIFT-SPM[18]	GIST[26] + BSub[27]	SIFT-SPM[18]+BSub[27]
-	KNN	76.30± 3.0	47.51±5.1	52.27±3.4	39.91±3.0
-	SVM	74.42± 3.0	68.69±3.6	50.98±3.6	48.91±3.0
CMA+GFK	KNN	**78.07±1.8**	49.84±5.5	52.97±2.7	39.08±2.6
CMA+GFK	SVM	**78.38± 3.1**	74.98±2.7	59.55±2.9	47.59±2.8
CMA+SA	KNN	**78.71±1.7**	54.08±6.2	51.33±4.2	38.21±2.6
CMA+SA	SVM	**78.49±3.1**	75.66±2.9	59.68±2.9	49.05±2.8

Table 1. Our method, CMA, improves performance independent of the feature choice for the scene classification task. Results here are shown with optimizing the unsupervised adaptation problem using either the geodesic flow kernel (GFK)[9] or the subspace alignment (SA) method [7]. Average precision (%) is recorded when training with 50 labeled images and testing on the immediately following 24 hours (480 images).

Figure 4. Sample human labeled images used for intersection traffic classification. Positive examples are shown in the top row (blue) and negative examples are shown in the bottom row (red).

Adaptation	Classifier	SIFT-SPM [18]	GIST [26]	DeCAF [3]
-	KNN	66.31± 0.6	72.77± 0.8	84.60± 0.7
-	SVM	79.26± 0.6	76.40± 0.7	85.92± 0.4
CMA+GFK	KNN	66.32± 0.2	72.60± 0.9	82.65± 0.5
CMA+GFK	SVM	80.24± 0.7	78.32± 0.6	**89.68± 0.1**
CMA+SA	KNN	65.06± 1.1	71.44± 1.3	81.97± 0.6
CMA+SA	SVM	79.79± 0.6	78.31± 0.7	**89.71± 0.1**

Table 3. Our algorithm improves performance on category recognition task. We evaluate our continuous manifold adaptation approach (CMA) on the task of labeling images of automobiles as either cars or trucks. We show results using two solutions to the unsupervised adaptation problem (GFK[9] and SA[7]) and two inner product based source classifiers (KNN and SVM). We compare across three types of features and demonstrate the benefit of using our algorithm for each feature choice, including a deep learning based feature that was tuned for object classification on all of ImageNet[3].

Figure 7. Our method clearly adapts to vehicle appearance as it evolves to look different from that in the labeled 50's training data. We show example images misclassified by non-adaptive SVM (DeCAF features) and correctly classified by CMA followed by the same SVM classifier. The 5 sedans and 5 trucks for which the SVM had the highest confidence (though incorrect) are displayed here.

Conclusions

- What does deep learning offer domain adaptation and vice-versa?
- Detection is a profitable “domain”
- Generalization beyond the two-domain setting is important

Domain adaptation and deep learning for large scale object recognition and detection

Prof. Trevor Darrell (UCB)

with

Judy Hoffman, Eric Tzeng, Jeff Donahue, Ross Girshick (UCB)

Kate Saenko (UML)