Domain Adaptation in a deep learning context

Tinne Tuytelaars

Work in collaboration with

  …

- T. Tommasi, N. Patricia, B. Caputo, T. Tuytelaars ***"A Deeper Look at Dataset Bias"*** GCPR 2015
- A. Raj, V. Namboodiri and T. Tuytelaars, ***"Subspace Alignment Based Domain Adaptation for RCNN Detector",*** BMVC 2015.

Some key questions

- How relevant is DA in combination with CNN features ?
- Is integration of DA in end-to-end learning the ultimate answer ?
- How do we properly evaluate DA methods ?
- Beyond classification ?

Some key questions

- How relevant is DA in combination with CNN features ?
- Is integration of DA in end-to-end learning the ultimate answer ?
- How do we properly evaluate DA methods ?
- Beyond classification ?

Domain Adaptation

- Domain
  = a set of data defined by its marginal and conditional distributions with respect to the assigned labels

- Dataset bias
  – Capture bias
  – Category or label bias
  – Negative bias

How relevant is DA with CNN ?

- Most widely used DA setting for CV:
  - Office Dataset
  - SURF + BoW representation

- Modern features (CNN) give better results even without DA

- “CNN features have been trained on very large datasets, to be robust to all possible types of variability. So we do not need any DA anymore.”

How relevant is DA with DNN?

- Always compare DA algorithms on top of the same representations

- Performance on target depends on:
  - performance on source
  - domain divergence
  - model that fits both source and target (e.g. negative bias, annotator bias)

How relevant is DA with DNN ?

- CNN features > SURF + BOW
  - More robust
  - Learned on larger datasets

- CNN features are data-driven
  - Be careful how to collect data, avoid unwanted biases
  - May make them more sensitive to domain shifts…

Some key questions

- How relevant is DA in combination with CNN features ?
- Is integration of DA in end-to-end learning the ultimate answer ?
- How do we properly evaluate DA methods ?
- Beyond classification ?

Integration of DA in DNN

- Learn representations that yield good classification AND are robust to domain changes, using deep / end-to-end learning

- “We can learn representations that are even more robust using DA”

Practical applications

- Use pretrained models of ImageNet on my own photo-collection
- Use off-the-shelve pedestrian detector even under low illumination conditions
- …

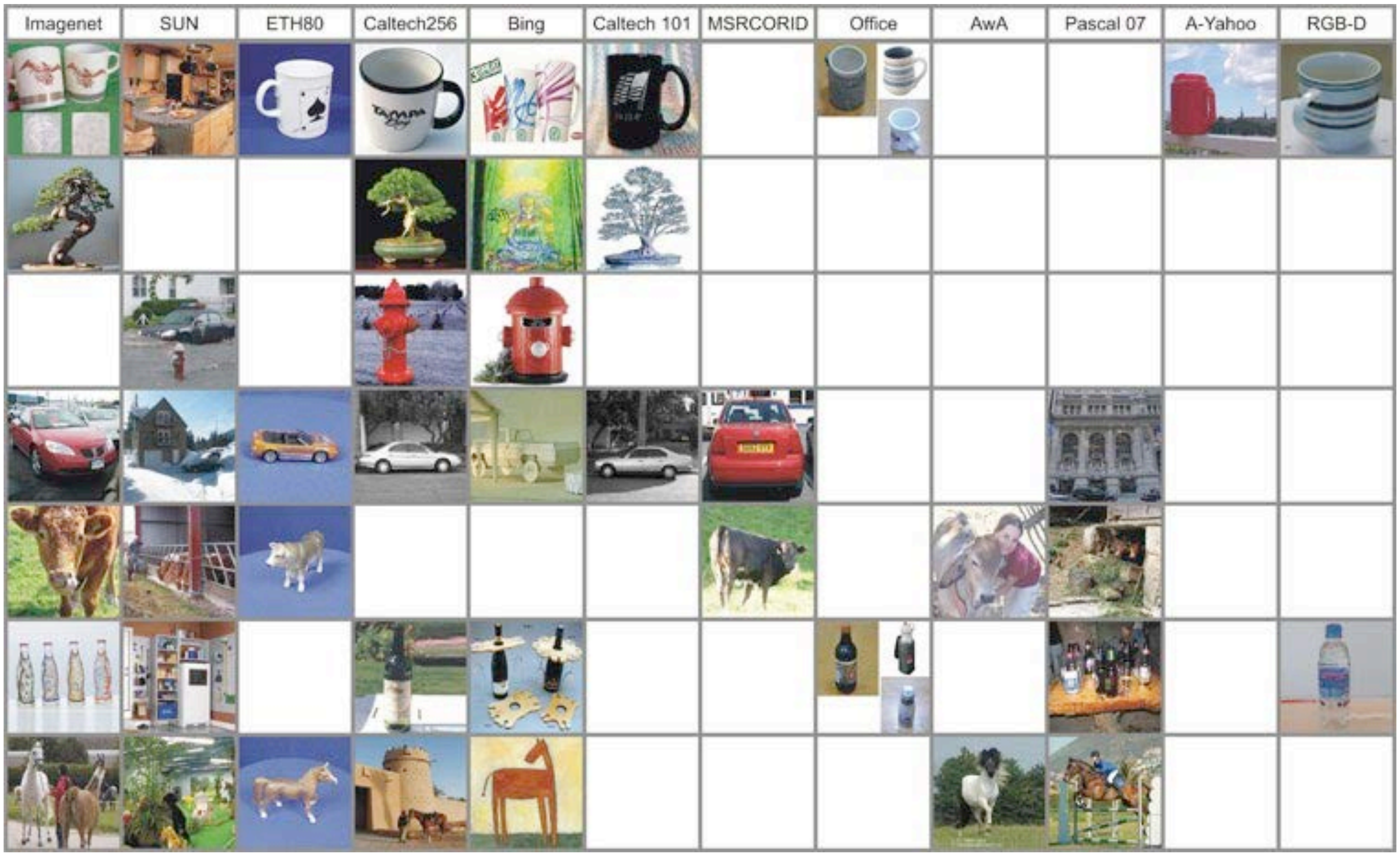
-> calls for simple, light adaptation methods

- Can we still use the simple DA methods developed before ?
  (in particular, we focus on subspace based methods)
- Are methods developed for BoW-features suitable for CNN features ?

Some key questions

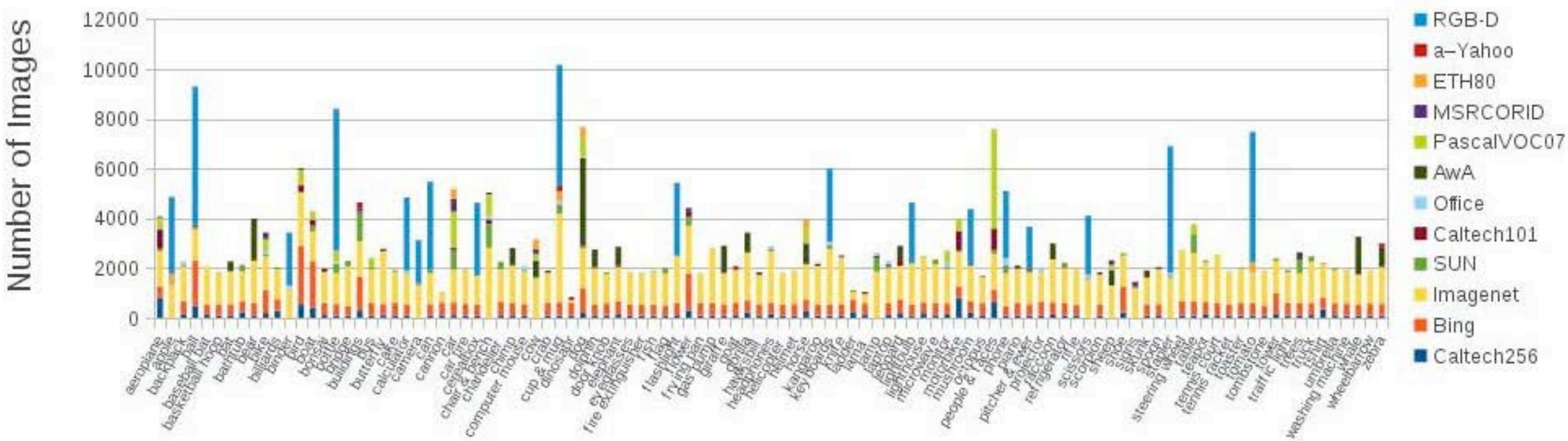
- How relevant is DA in combination with CNN features ?
- Is integration of DA in end-to-end learning the ultimate answer ?
- How do we properly evaluate DA methods ?
- Beyond classification ?

A cross-dataset benchmark



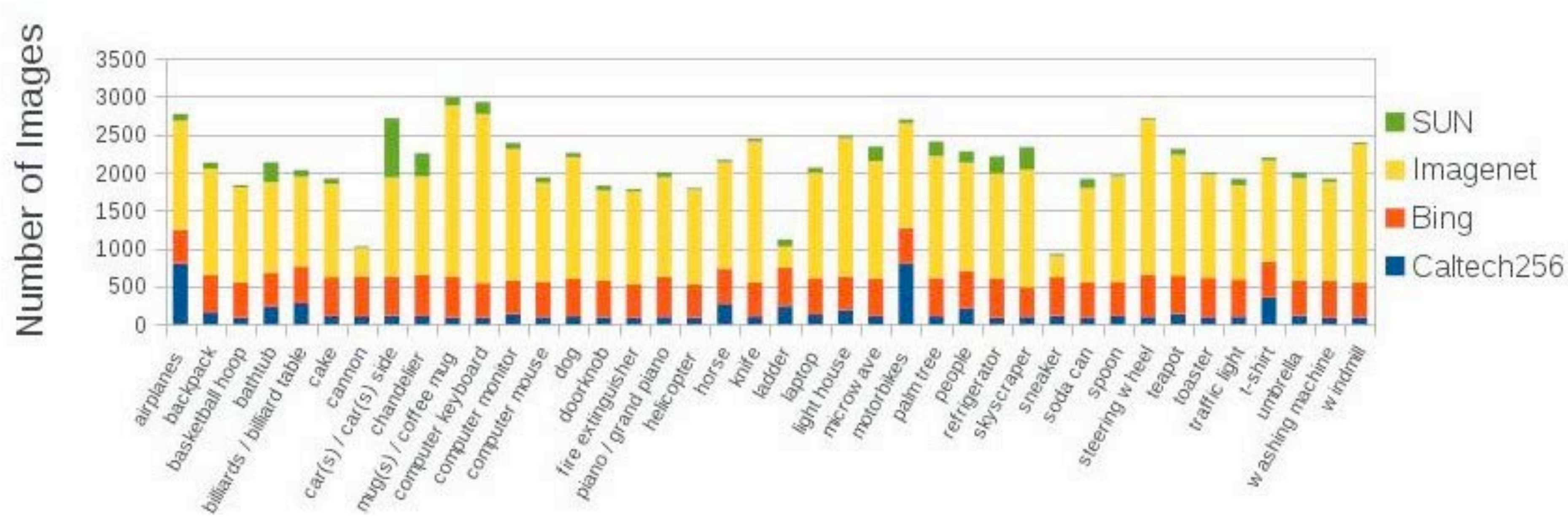
Cross-dataset benchmark

- Sparse Set



Cross-dataset benchmark

- Dense set



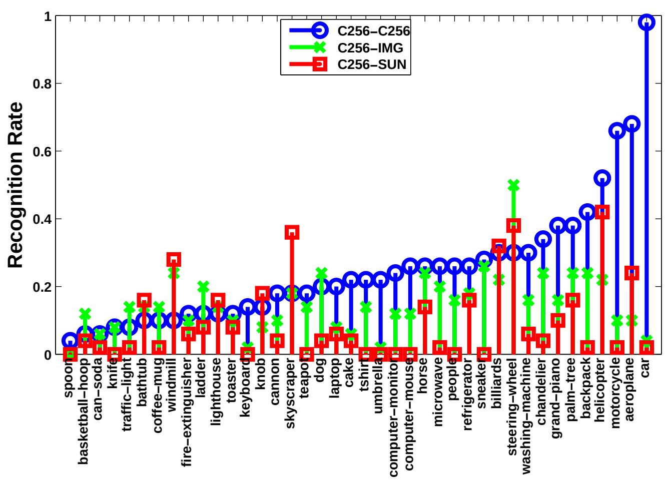
New evaluation criterion

- % drop
- CD measure

$$CD = \frac{1}{1 + exp^{-\{(Self - Mean\ \ Others)/100\}}} \ .$$

Some experimental results

BOWsift



DeCAF7

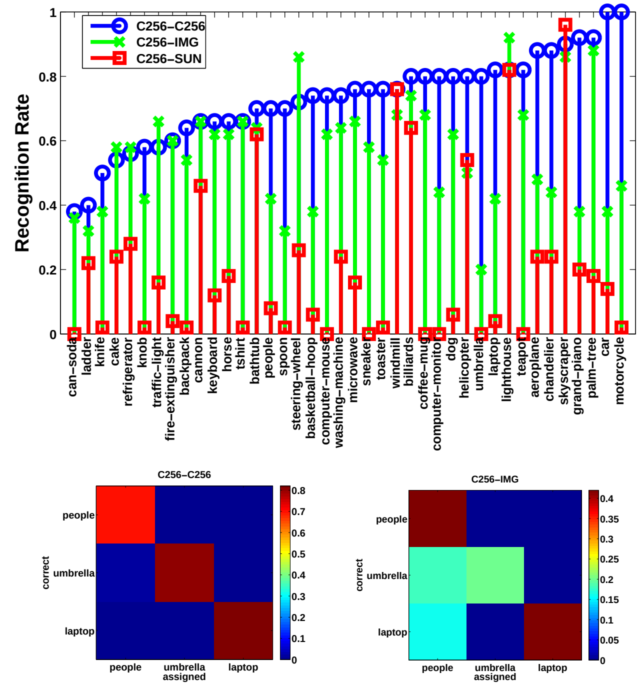


Table 3 Recognition rate per class from the multiclass cross-dataset generalization test. C256, IMG and SUN stand respectively for Caltech256, Imagenet and SUN datasets. We indicate with "train-test" the pair of datasets used in training and testing.

Some experimental results

Car

Feature	train \ test	P	S	E	M	% Drop	CD
BOWsift	P	28.6	26.4	26.7	29.2	3.9	0.50
	S	25.4	26.8	25.7	25.7	4.3	0.50
	E	14.7	9.1	98.5	25.2	83.4	**0.69**
	M	17.4	8.7	9.7	90.5	86.8	**0.69**
DeCAF6	P	26.2	19.2	44.3	42.6	-35.1	0.47
	S	15.4	15.6	19.9	17.9	-13.9	0.49
	E	27.3	18.7	100.0	93.5	53.5	**0.63**
	M	41.6	12.8	95.8	99.7	49.8	**0.62**
DeCAF7	P	14.5	11.8	28.9	28.3	-59.1	0.47
	S	27.1	27.4	32.8	28.1	-6.8	0.49
	E	31.0	24.0	100.0	90.9	51.3	**0.62**
	M	43.8	14.9	93.8	99.7	49.0	**0.62**

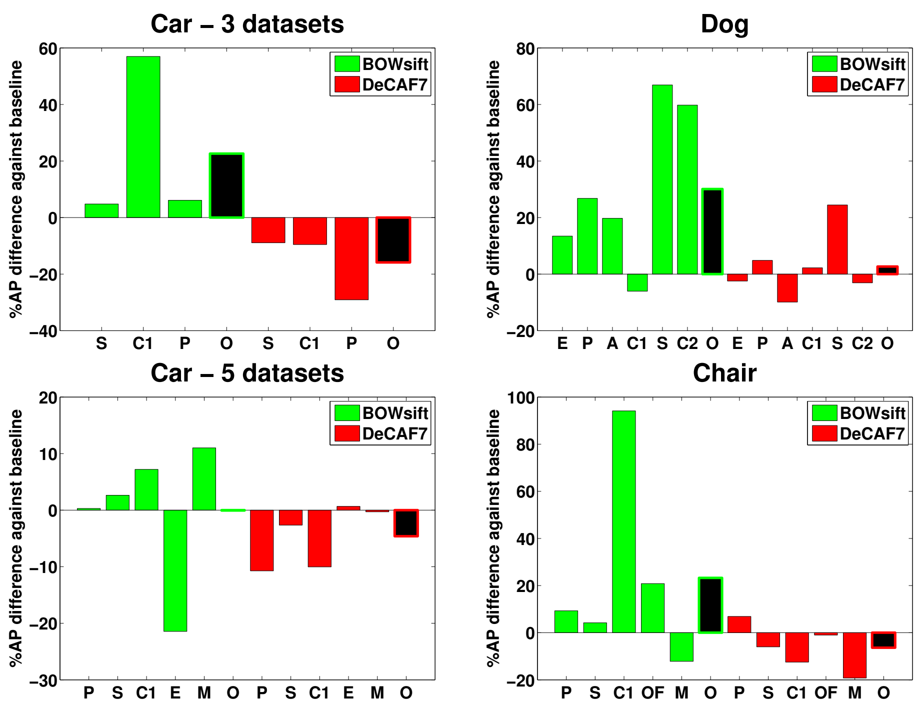
Cow

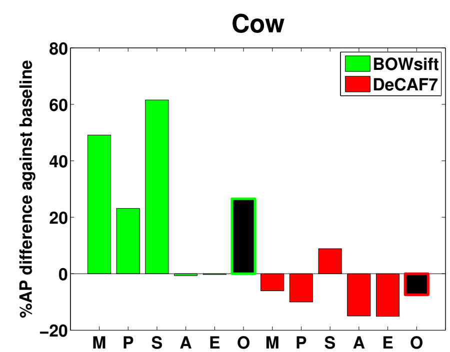
Feature	train \ test	P	S	E	M	% Drop	CD
BOWsift	P	14.1	18.8	11.9	17.9	-15.1	0.49
	S	16.1	38.3	12.5	27.1	51.4	**0.54**
	E	2.3	2.3	29.8	2.0	92.6	**0.57**
	M	8.6	17.4	2.4	52.5	82.0	**0.61**
DeCAF6	P	35.3	34.4	20.2	39.1	11.4	**0.51**
	S	18.6	62.7	10.7	33.5	66.7	**0.60**
	E	4.4	8.3	91.9	4.1	93.9	**0.70**
	M	13.8	46.7	9.3	97.2	76.0	**0.68**
DeCAF7	P	46.6	44.0	31.3	47.3	12.3	**0.51**
	S	12.8	40.1	12.1	23.6	59.7	**0.56**
	E	7.2	7.7	90.1	5.3	92.5	**0.70**
	M	14.1	39.4	10.4	97.7	78.2	**0.68**

Cow - fixed negatives

Feature	train \ test	P	S	E	M	% Drop	CD
BOWsift	P	14.1	16.2	12.0	18.6	-10.7	0.49
	S	18.7	38.3	17.0	35.8	-37.8	**0.53**
	E	23.5	22.2	29.8	14.0	33.3	**0.52**
	M	8.9	9.4	2.0	52.5	87.1	**0.61**
DeCAF6	P	35.3	32.7	16.9	46.6	9.1	0.50
	S	38.5	62.7	21.0	69.5	31.4	**0.54**
	E	6.1	7.9	91.9	4.6	93.2	**0.70**
	M	65.2	62.6	51.5	97.2	38.5	**0.59**
DeCAF7	P	46.6	39.1	26.4	48.9	18.2	**0.52**
	S	25.7	40.1	12.8	43.3	31.9	**0.53**
	E	11.2	9.0	90.1	11.0	88.4	**0.69**
	M	69.4	55.1	47.4	97.7	41.3	**0.59**

Undoing dataset bias

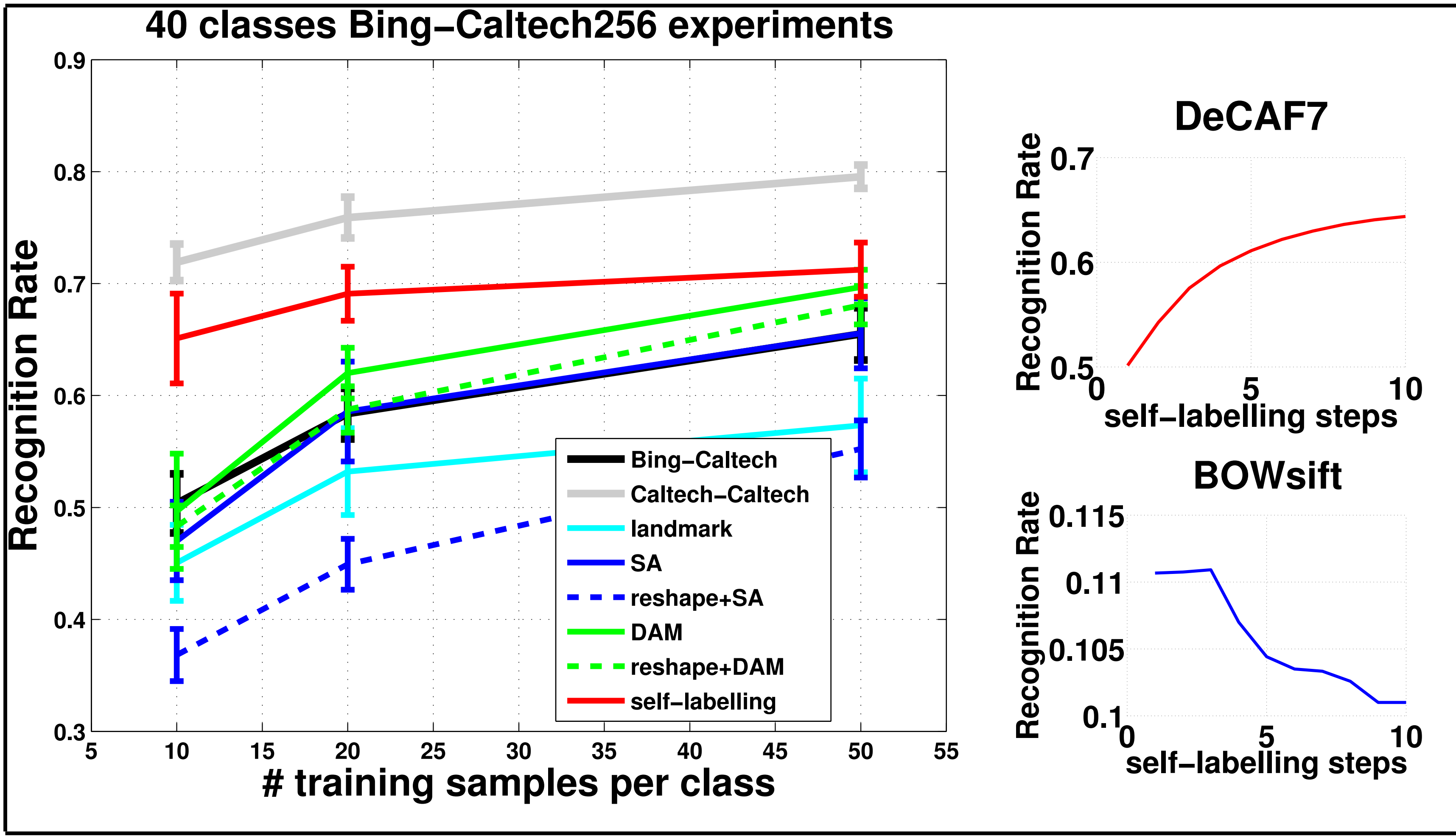




Cow	*Self*	*Other*
M	98.97	86.56
P	53.67	34.29
S	40.89	29.20
A	44.76	25.94
E	99.40	86.56

Fig. 2 Percentage difference in average precision between the results of *Unbias* and the baseline *All* over each target dataset. P,S,E,M,A,C1,C2,OF stand respectively for the datasets Pascal VOC07, SUN, ETH80, MSRCORID, AwA, Caltech101, Caltech256 and Office. With O we indicate the overall value, *i.e.* the average of the percentage difference over all the considered datasets (shown in black).

DA methods with DeCAF7

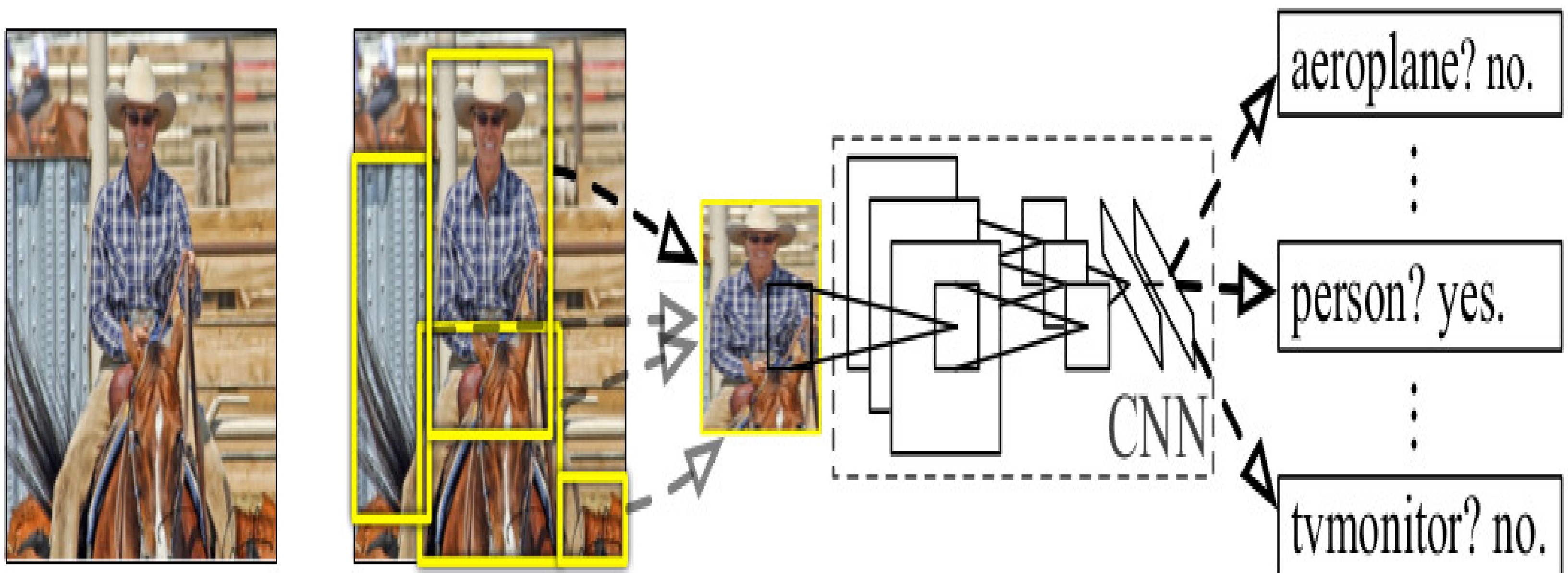


Conclusions

- CNN features (DeCaf7) are more powerful, but this by itself doesn't suffice to avoid domain shift
- Sometimes too specific and even worse performance (both class – and dataset dependent)
- Negative bias problem remains
- Standard DA methods do not always perform well on CNN-features – more difficult to generalize ?
- Best results with self-labeling on target data

Adapting R-CNN detector from Pascal VOC to Microsoft COCO

(a) PASCAL (b) PASCAL (c) PASCAL (d) COCO (e) COCO (f) COCO



- Initialize the detection on Target dataset
- Avoid non maximum suppression while learning target subspace
- Discard noisy detections while initialization
- Initial RCNN is trained on Pascal VOC 2012
- We generate class specific target subspaces and apply subspace alignment approach on each class separately

No.	class	RCNN- No Transform	RCNN - Full Transform	Proposed	DPM
1	plane	36.72	35.44	**40.1**	35.1
2	bicycle	21.26	18.95	**23.28**	1.9
3	bird	12.50	12.37	**13.63**	3.7
4	boat	10.45	8.8	**10.61**	2.3
5	bottle	8.75	**11.46**	8.11	7
6	bus	37.47	38.12	40.64	**45.4**
7	car	20.6	20.4	**22.5**	18.3
8	cat	42.4	43.6	**45.6**	8.6
9	chair	**9.6**	6.3	8.8	6.3
10	cow	23.28	20.40	**25.3**	17
11	table	15.9	14.9	**17.3**	4.8
12	dog	28.42	**32.72**	31.3	5.8
13	horse	30.7	31.11	32.9	**35.3**

No.	class	RCNN- No Transform	RCNN - Full Transform	Proposed	DPM
14	motorbike	31.2	29.05	**34.6**	25.4
15	person	27.8	28.8	**30.9**	17.5
16	plant	12.65	7.34	**13.7**	4.1
17	sheep	19.99	21.04	**22.4**	14.5
18	sofa	14.6	8.4	**15.5**	9.6
19	train	39.2	38.4	**41.64**	31.7
20	tv	28.6	26.4	**29.9**	27.9
	Mean AP	23.60	22.7	**25.43**	16.9

Questions ?