A deeper look at dataset bias

A Deeper Look at Dataset Bias

Tatiana Tommasi[1⋆], Novi Patricia[2], Barbara Caputo[3], Tinne Tuytelaars[4⋆⋆]

[1]Department of Computer Science, University of North Carolina, Chapel Hill, USA
[2]Idiap Research Institute, Martigny; EPFL, Lausanne, Switzerland
[3]University of Rome, La Sapienza, Italy
[4]KU Leuven, ESAT-PSI, iMinds, Belgium

Abstract. The presence of a bias in each image data collection has recently attracted a lot of attention in the computer vision community showing the limits in generalization of any learning method trained on a specific dataset. At the same time, with the rapid development of deep learning architectures, the activation values of Convolutional Neural Networks (CNN) are emerging as reliable and robust image descriptors. In this paper we propose to verify the potential of the DeCAF features when facing the dataset bias problem. We conduct a series of analyses looking at how existing datasets differ among each other and verifying the performance of existing debiasing methods under different representations. We learn important lessons on which part of the dataset bias problem can be considered solved and which open questions still need to be tackled.

1 Introduction

Since its spectacular success in the 2012 edition of the Imagenet Large Scale Visual Recognition Challenge (ILSVRC, [28]), deep learning has dramatically changed the research landscape in visual recognition [20]. By training a Convolutional Neural Network (CNN) over millions of data it is possible to get impressively high quality object annotations [1] and detections [38]. A large number of studies have recently proposed improvements over the CNN architecture of Krizhevsky *et al.* [20] with the aim to better suit an ever increasing typology of visual applications [16,38,30]. At the same time, the activation values of the final hidden layers have quickly gained the status of off-the-shelf state of the art features [27]. Indeed, several works demonstrated that DeCAF (as well as Caffe [6], Overfeat [32], VGG-CNN [3] and other implementations) can be used as powerful image descriptors [3,14]. The improvements obtained over previous methods are so impressive that one might wonder whether they can be considered as a sort of "universal features", *i.e.* image descriptors that can be helpful in any possible visual recognition problem. The aim of this paper is to contribute to answering this question when focusing on the bias of existing computer vision datasets.

The main causes and consequences of the *dataset bias* have been pointed out and named in [34]. The *capture bias* is related to how the images are acquired both in terms of the used device and of the collector preferences for point of view, lighting conditions,

⋆ Work done mainly while at KU Leuven, Belgium.

⋆⋆ T. Tommasi and T. Tuytelaars acknowledge the support of the FP7 EC project AXES and of the FP7 ERC Starting Grant 240530 COGNIMUND.

etc. The *category or label bias* is due to a poor definition of the visual semantic categories and to the in-class variability: similar images may be annotated with different names and the same name can be assigned to visually different images. Finally, each collection may contain a distinct set of categories and this causes the *negative bias*. If we focus only on the classes shared among them, the rest of the world will be defined differently depending on the collection. All these bias aspects induce a generalization problem when training and testing a learning algorithm on images extracted from different collections. Previous work seemed to imply that this issue was solved, or on the way to be solved, by using CNN features [6,37]. However, the evaluation is generally restricted to controlled cases limited to specific visual domain shift [6,18] or with images extracted from the testing collection available at training time [37,26].

In this work we revisit and scale up the dataset bias analysis, making two contributions:

1. we asses the performance of the DeCAF CNN features on the most comprehensive experimental setup existing for dataset bias. We build on the setting proposed in [33], consisting of a cross-dataset testbed over twelve different databases.
2. we propose a new measure to quantify the ability of a given algorithm to address the dataset bias. As opposed to what was proposed in [34], our measure takes into account both the performance obtained on the in-dataset task and the percentage drop in performance across datasets.

Our experiments evaluate the suitability of CNN features for attacking the dataset bias problem, pointing out that: (1) the capture bias is class-dependent and can be enhanced by the CNN representation due to the influence of the classes on which the neural network was originally trained; (2) the negative bias persists regardless of the representation; (3) attempts of undoing the dataset bias with existing ad-hoc learning algorithms do not help, while some previously discarded adaptive strategies appear effective; (4) fine-tuning the CNN network does not fit in the dataset bias setting and if naïvely forced does not seem beneficial.

The picture emerging from these findings is that of a problem open for research and in need for new directions, able to accommodate at the same time the potential of deep learning and the difficulties of large scale cross-database generalization.

2 Evaluation Protocol

We describe here the setup adopted for the experiments and we introduce the measures used to evaluate the cross-dataset generalization performance.

Datasets & Features. We focus on twelve datasets, created and used before for object categorization, that have been recently organized in a cross-dataset testbed with the definition of two data setups [33]:

– **sparse set**. It contains 105 Imagenet classes [5] aligned to 95 classes of Caltech256 [15] and Bing [35], 89 classes of SUN [36], 35 classes of Caltech101 [10], 17 classes of Office [31], 18 classes of RGB-D [21], 16 classes of Animals with Attributes (AwA) [22] and Pascal VOC07 [8], 13 classes of MSRCORID [25], 7 classes of ETH80 [23], and 4 classes of a-Yahoo [9].

– **dense set**. It contains 40 classes shared by Bing, Caltech256, Imagenet and SUN.

The testbed has been released together with three feature representations:

– **BOWsift**: dense SIFT descriptors [24] extracted with the protocol defined for the ILSVRC2010 contest [29] and quantized into a BOW representation based on a vocabulary of 1000 visual words;
– **DeCAF6, DeCAF7**: the mean-centered raw RGB pixel intensity values of all the collection images (warped to 256x256) are given as input to the CNN architecture of Krizhevsky *et al.* by using the DeCAF implementation [6]. The activation values of the 4096 neurons in the 6-th and 7-th layers of the network are considered as image descriptors.

In our experiments we use the L2-normalized version of the feature vectors and adopt the z-score normalization for the BOWsift features when testing domain adaptation methods. We mostly focus on the results obtained with the DeCAF features and use the BOWsift representation as a reference baseline.

Evaluation Measures. We analyze both the *in-dataset* (training and testing on samples extracted from the same dataset) and the *cross-dataset* (training and testing samples belonging to different collections) performance. We use $Self$ to specify the in-dataset performance and $Mean\ Other$ for the average cross-dataset performance over multiple test collections.

In [34] cross dataset generalization was evaluated through the percentage drop ($\%\ Drop$) between $Self$ and $Mean\ Others$. However, being a relative measure, it loses the information on the value of $Self$ which is important if we want to compare the effect of different learning methods or different representations. For instance a 75% drop w.r.t a 100% self average precision has a different meaning than a 75% drop w.r.t. a 25% self average precision. To overcome this drawback, we propose here a different *Cross-Dataset (CD)* measure defined as

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

CD uses directly the difference $(Self - Mean\ \ Others)$ while the sigmoid function rescales this value between 0 and 1. This allows for the comparison among the results of experiments with different setups. Specifically CD values over 0.5 indicate a presence of a bias, which becomes more significant as CD gets close to 1. On the other hand, CD values below 0.5 correspond to cases where either $Mean\ Other \geq Self$ or the $Self$ result is very low. Both these conditions indicate that the learned model is not reliable on the data of its own collection and it is difficult to draw any conclusion from its cross-dataset performance.

3 Studying the Sparse set

Dataset Recognition. One of the effect of the capture bias is that it makes any dataset easily recognizable. We want to evaluate whether this effect is enhanced or decreased

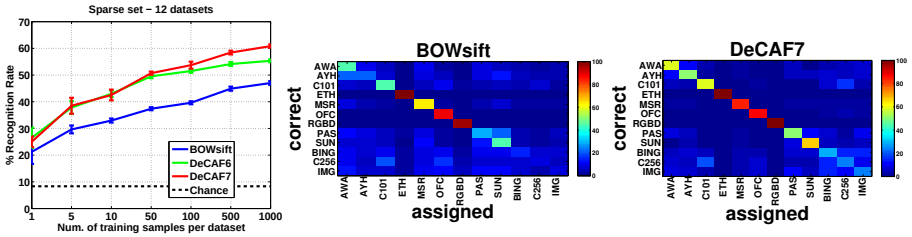


Fig. 1. Name the dataset experiment over the sparse setup with 12 datasets. The title of each confusion matrix indicates the feature used for the corresponding experiments.

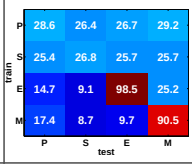
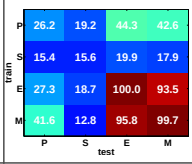
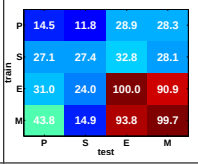
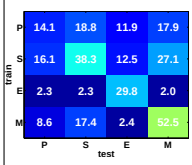
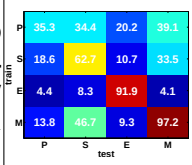
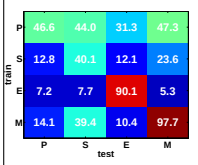
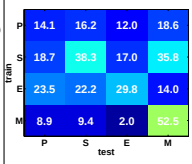
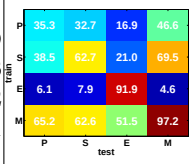
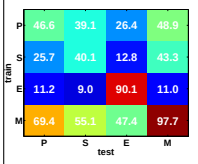
by the use of the CNN features. To do it we run the *name the dataset* test [34] on the sparse data setup. We extract randomly 1000 images from each of the 12 collections and we train a 12-way linear SVM classifier that we then test on a disjoint set of 300 images. The experiment is repeated 10 times with different data splits and we report the obtained average results in Figure 1. The plot on the left indicates that DeCAF allows for a much better separation among the collections than what is obtained with BOWsift. In particular DeCAF7 shows an advantage over DeCAF6 for large number of training samples. From the confusion matrices (middle and right in Figure 1) we see that it is easy to distinguish ETH80, Office and RGB-D datasets from all the others regardless of the used representation, given the specific lab-nature of these collections. DeCAF captures better than BOWsift the characteristics of A-Yahoo, MSRCORID, Pascal VOC07 and SUN, improving the recognition results on them. Finally, Bing, Caltech256 and Imagenet are the datasets with the highest confusion level, an effect mainly due to the large number of classes and images per class. Still, this confusion decreases when using DeCAF.

These experiments show that the idiosyncrasies of each data collection become more evident when using a highly accurate representation. However, the dataset recognition performance does not provide an insight on how the classes in each collection are related among each other, nor how a specific class model will generalize to other datasets. We look into this problem in the following paragraph.

Class-Specific cross-dataset generalization test. We study the effect of the CNN features on the cross-dataset performance of two object class models: *car* and *cow*. Four collections in the sparse set contain images labeled with these object classes: PascalVOC07 (P), SUN (S), ETH80 (E), and MSRCORID (M). For the class car we selected randomly from each dataset two groups of 50 positive/1000 negative examples respectively for training and testing. For the class cow we considered 30 positive/1000 negative examples in training and 18 positive/1000 negative examples in testing. We repeat the sample selection 10 times and the average precision results obtained by linear SVM are presented in Table 1.

Coherently with what deduced over all the classes from the *name the dataset* experiment, scene-centric (P,S) and object-centric (E,M) collections appear separated among each other. For the first ones, the low in-dataset results are mainly due to their multi-label nature: an image labeled as people may still contain a car and this creates confusion both at training and at test time. The final effect is a cross-dataset performance

Table 1. Binary cross-dataset generalization for two example categories, car and cow. Each matrix contains the object classification performance (AP) when training on one dataset (rows) and testing on another (columns). The diagonal elements correspond to the self results, *i.e.* training and testing on the same dataset. We report in bold the CD values higher than 0.5.

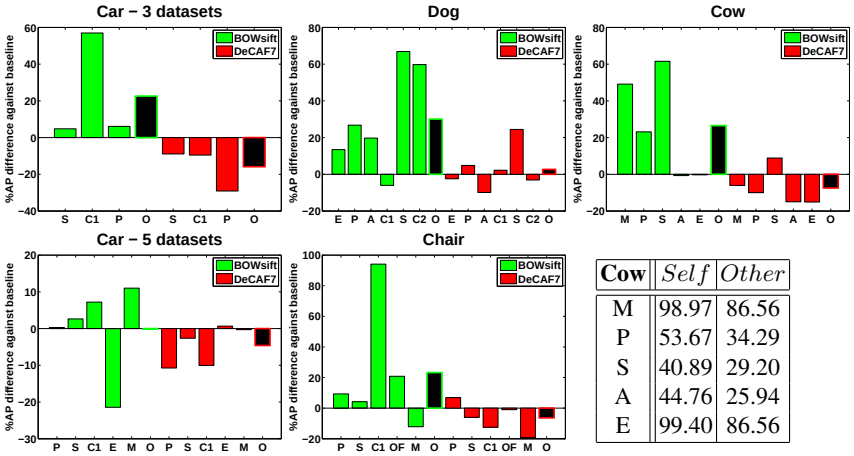
	BOWsift	% Drop	CD	DeCAF6	% Drop	CD	DeCAF7	% Drop	CD
Car		3.9	0.50		-35.1	0.47		-59.1	0.47
		4.3	0.50		-13.9	0.49		-6.8	0.49
		83.4	**0.69**		53.5	**0.63**		51.3	**0.62**
		86.8	**0.69**		49.8	**0.62**		49.0	**0.62**
Cow		-15.1	0.49		11.4	**0.51**		12.3	**0.51**
		51.4	**0.54**		66.7	**0.60**		59.7	**0.56**
		92.6	**0.57**		93.9	**0.70**		92.5	**0.70**
		82.0	**0.61**		76.0	**0.68**		78.2	**0.68**
Cow - fixed negatives		-10.7	0.49		9.1	0.50		18.2	**0.52**
		-37.8	**0.53**		31.4	**0.54**		31.9	**0.53**
		33.3	**0.52**		93.2	**0.70**		88.4	**0.69**
		87.1	**0.61**		38.5	**0.59**		41.3	**0.59**

higher than the respective in-dataset one. This behavior becomes even more evident when using DeCAF than with BOWsift.

Although the *name the dataset* experiment indicated almost no overall confusion between E and M, the per-class results on car and cow show different trends. Learning a *car* model from images of toys (E) or of real objects (M) does not seem so different in terms of the final testing performance when using DeCAF. The diagonal matrix values prominent with BOWsift are surrounded by high average precision results for DeCAF. On the other hand, recognizing a living non-rigid object like a *cow* is more challenging. An important factor that may influence these results is the high level nature of the DeCAF representation: they are obtained as a byproduct of a training process over 1000 object classes [6] which cover several vehicles and animal categories. The class *car* is in this set, but *cow* is not. This intrinsically induce a category-specific bias effect, which may augment the image collection differences. Overall the DeCAF features provide a high performance inside each collection, but the difference between the in-dataset and cross-dataset results remains large almost as with BOWsift.

We also re-run the experiments on the class cow by using a fixed negative set in the test always extracted from the training collection. The visible increase in the cross-dataset results indicate that the negative set bias maintain its effect regardless of the used representation.

From the values of $\%Drop$ and CD we see that these two measures may have a different behavior: for the class cow with BOWsift, the $\%Drop$ value for E (92.6) is



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 Pascal VOC07, SUN, ETH80, MSRCORID, AwA, Caltech101, Caltech256 and Office. With O (in black) we indicate the overall value: average percentage difference over all the considered datasets.

higher than the corresponding value for M (82.0), but the opposite happens for CD (respectively 0.57 and 0.61). The reason is that CD integrates the information on the in-dataset recognition which is higher and more reliable for M. Passing from BOWsift to DeCAF the CD value increases in some cases indicating a more significant bias.

On the basis of the presented results we can state that the DeCAF features are not fully solving the dataset bias. Although similar conclusions have been mentioned in a previous publication [18], our more extensive analysis provides a reliable measure to evaluate the bias and explicitly indicate some of the main causes of the observed effect: (1) the capture bias appears class-dependent and may be influenced by the original classes on which the CNN features have been trained; (2) the negative bias persists regardless of the feature used to represent the data.

Undoing the Dataset Bias. We focus here on the method proposed in [19] to overcome the dataset bias. Our aim is to verify its effect when using the DeCAF features. The *Unbias* approach has a formulation similar to multi-task learning: the available images of multiple datasets are kept separated as belonging to different tasks and a max-margin model is learned from the information shared over all of them. We run the experiments focusing on the classes *car*, *cow*, *dog* and *chair*, reproducing a similar setup to what previously used in [19] and using the original implementation of the *Unbias* method provided by the authors. For the class car we consider two settings with three and five datasets, while we use five datasets for cow and chair and six datasets for dog. One of the datasets is left out in round for testing while all the others are used as sources of training samples[1].

[1] More details about the method and the experimental setup can be found in the supplementary material.

We compare the obtained results against those produced by a linear SVM when *All* the training images of the source datasets are considered together. We show the percentage relative difference in terms of average precision for these two learning strategies in Figure 2. The results indicate that, in most cases when using BOWsift the *Unbias* method improves over the plain *All* SVM, while the opposite happens when using DeCAF7. As already suggested by the results of the cross-dataset generalization test, the DeCAF features, by capturing the image details, may enhance the differences among the same object category in different collections. As a consequence, the amount of shared information among the collections decreases, together with the effectiveness of the methods that leverage over it. On the other hand, removing the dataset separation and considering all the images together provides a better coverage of the object variability and allows for a higher cross-dataset performance.

In the last column of Figure 2 we present the results obtained with the class *cow* together with the average precision per dataset when using DeCAF7. The table allows to compare the performance of training and testing on the same dataset ($Self$) against the best result between *Unbias* and *All* (indicated as $Other$). Despite the good performance obtained by directly learning on other datasets, the obtained results are still lower than what can be expected having access to training samples of each collection. This suggests that an adaptation process from generic to specific is still necessary to close the gap. Similar trends can be observed for the other categories.

4 Studying the Dense set

Dataset Recognition. A second group of experiments on the dense setup allows us to analyze the differences among the datasets avoiding the negative set bias. We run again the *name the dataset* test maintaining the balance among the 40 classes shared by Caltech256, Bing, SUN and Imagenet. We consider a set of 5 samples per object class in testing and an increasing amount of training samples per class from 1 to 15. The results in Figure 3 indicate again the better performance of DeCAF7 over DeCAF6 and BOWsift. From the confusion matrices it is clear that the separation between object- (Bing, Caltech256, Imagenet) and scene-centric (SUN) datasets is quite easy regardless of the representation, while the differences among the object-centric collections become more evident when passing from BOW to DeCAF.

Since all the datasets contain the same object classes, we are in fact reproducing a setup generally adopted for domain adaptation [13,11]. By identifying each dataset with a domain, we can interpret the results of this experiment as an indication of the domain divergence [2] and deduce that a model trained on SUN will perform poorly on the object-centric collections and vice versa. On the other hand, a better cross dataset generalization should be observed among Imagenet, Caltech256 and Bing. We verify it in the following sections.

Cross-dataset generalization test. We consider the same setup used before with 15 samples per class from each collection in training and 5 samples per class in test. However, now we train a one-vs-all multiclass SVM per dataset. Due to its noisy nature we exclude Bing here and we dedicate more attention to it in the next paragraph.

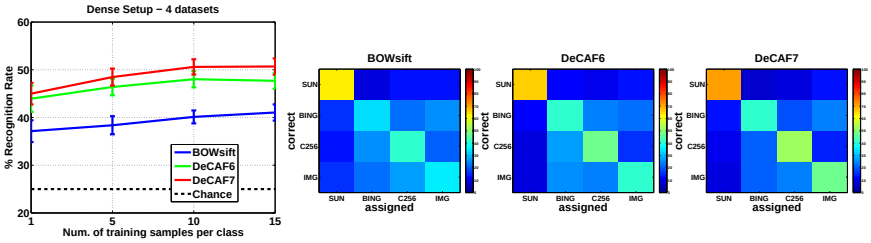
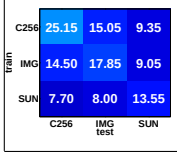
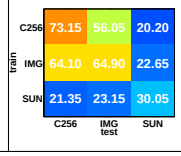


Fig. 3. Name the dataset experiment over the dense setup with 4 datasets. The title of each confusion matrix indicates the feature used for the corresponding experiments.

Table 2. Multiclass cross-dataset generalization performance (recognition rate). The percentage difference between the self results and the average of the other results per row correspond to the value indicated in the column $\%\ Drop$. CD is our newly proposed cross-dataset measure.

BOWsift	% Drop	CD	DeCAF7	% Drop	CD
	51.5	0.53		47.9	0.58
	34.0	0.52		33.2	0.55
	42.1	0.51		25.9	0.52

The average recognition rate results over 10 data splits are reported in Table 2. By comparing the values of $\%Drop$ and CD we observe that they provide opposite messages. The first suggests that we get a better generalization when passing from BOWsift to DeCAF7. However, considering the higher $Self$ result, CD evaluates the dataset bias as more significant when using DeCAF7. The expectation indicated before on the cross-dataset performance are confirmed here: the classification models learned on Caltech256 and Imagenet have low recognition rate on SUN. Generalizing between Caltech256 and Imagenet, instead, appears easier and the results show a particular behavior: although the classifier on Caltech256 tends to fail more on Imagenet than on itself, when training on Imagenet the in-dataset and cross-dataset performance are almost the same. Of course we have to remind that the DeCAF features were defined over Imagenet samples and this can be part of the cause of the observed asymmetric results.

Noisy Source Data and Domain Adaptation. Until now we have discussed and demonstrated empirically that the difference among two data collections can originate from multiple and often co-occurring causes. However the standard assumption is that the label assigned to each image is correct. In some practical cases this condition does not hold, as in learning from web data [4]. Some state-of-art domain adaptation methods seem perfectly suited for this task (see Figure 4 top part) and we use them here to evaluate the cross-dataset generalization performance when training on Bing (noisy object-centric source domain) and testing on Caltech256 and SUN (respectively an object-centric and a scene-centric target domain).

The obtained results go in the same direction of what was observed previously with the *Unbias* method. Despite the presence of noisy data, selecting them (landmark) or grouping the samples (reshape+SA, reshape+DAM) do not seem to work better than

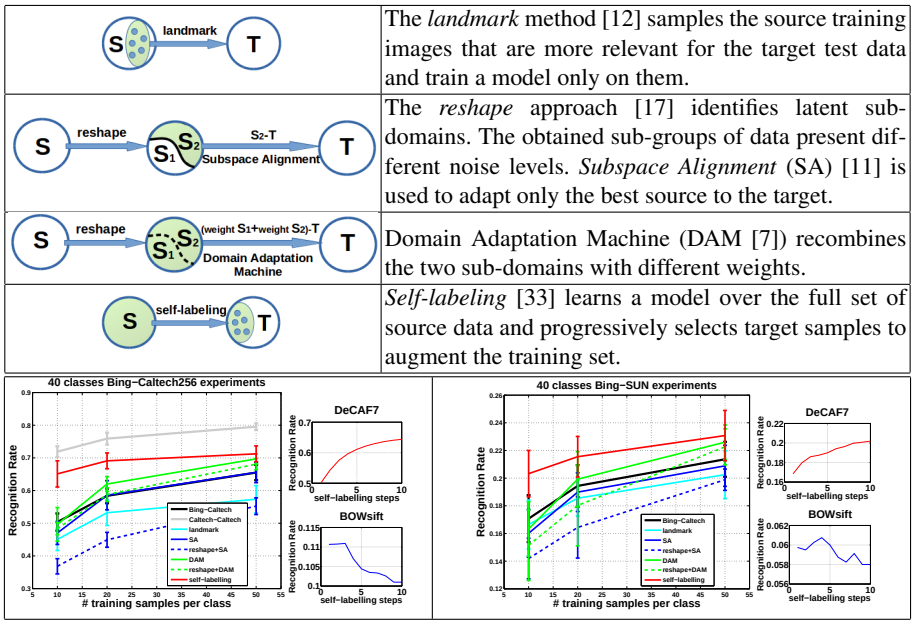


Fig. 4. Top: schematic description of the used domain adaptation methods. Bottom: Results of the Bing-Caltech256 and Bing-SUN experiments with DeCAF7. We report the performance of different domain adaptation methods (big plots) together with the recognition rate obtained in 10 subsequent steps of the self-labeling procedure (small plots). For the last ones we show the performance obtained both with DeCAF7 and and with BOWsift when having originally 10 samples per class from Bing.

just using all the source data at once. On the other hand, keeping all the source data together and augmenting them with target samples by *self-labeling* [33] consistently improves the original results. One well known drawback of this strategy is that progressively accumulated errors in the target annotations may lead to significant drift from the correct solution. However, when working with DeCAF features this risk appears highly reduced as can be appreciated by looking at the recognition rate obtained over ten iterations of the target selection procedure and considering the comparison against BOWsift (small plots in Figure 4).

Fine-Tuning. As indicated in section 2 the DeCAF CNN features were obtained from an initial pre-trained network whose parameters remain untouched. Fine-tuning the network before using it for recognition on a new task is an alternative strategy which demonstrated good results in transfer learning [26,37]. To complete our analysis we clarify here that this fine-tuning process does not fit in the dataset bias setting.

A network pre-trained on a dataset D is generally fine-tuned on a new dataset D' when the final task is also tested on D'. Thus the scheme (train,fine-tune,test) corresponds to (D, D', D'). For dataset bias, the condition is instead (D, D', D''): here D' and D'' are different collection and no labeled data from D'' is available at training time. The advantage of fine-tuning consists in making the network spe-

cific for D' [3], which in our setting can worsen the bias with respect to D''. By using the Caffe CNN implementation we fine-tuned the Imagenet (D) pre-trained network on the dense set, specifically on Caltech256 (5046 train images) and SUN (3015 train images), reserving respectively 1500 and 1300 images as test samples. The in-dataset and cross-dataset experimental results are: $(Caltech256(D'), Caltech256(D')) = 86.4\%$; $(Caltech256(D'), SUN(D'')) = 25.7\%$; $(SUN(D'), Caltech256(D'')) = 37.5\%$; $(SUN(D'), SUN(D')) = 41.1\%$. Compared with what presented in Table 2 these results show the advantage of fine-tuning in terms of in-dataset recognition rate. However they also indicate that the fine-tuning process does not remove the cross-dataset bias ($86.4\% > 25.7\%; 41.1\% > 37.5\%$) and that using the wrong dataset to refine the network can be detrimental ($86.4\% > 37.5\%; 41.1\% > 25.7\%$).

5 Conclusions

In this paper we attempted at positioning the dataset bias problem in the CNN-based features arena with an extensive experimental evaluation. At the same time, we pushed the envelope in terms of the scale and complexity of the evaluation protocol, so to be able to analyze all the different nuances of the problem. We focused on DeCAF features, as they are popular CNN-learned descriptors, and for the impressive results obtained so far in several visual recognition domains.

A first main result of our analysis is that DeCAF not only does not solve the dataset bias problem in general, but in some cases (both class- and dataset-dependent) they capture specific information that, although otherwise useful, induce a low performance in the cross-dataset object categorization task. The high level nature of the CNN features add a further hidden bias that needs to be considered when comparing the experimental results against standard hand-crafted representations. Moreover, the negative bias remains, as it cannot intrinsically be removed (or alleviated) by changing feature representation. A second result concerns the effectiveness of learning methods applied over the chosen features: nor a method specifically designed to undo the dataset bias, neither algorithms successfully used in the domain adaptation setting seem to work when applied over DeCAF features. It appears as if the highly descriptive power of the features, that determined much of their successes so far, in the particular dataset-bias setting backfires, as it makes the task of learning how to extract general information across different data collection more difficult. Interestingly, a simple selection procedure based on target self-labeling leads to a significant increase in performance. Finally, a third outcome derives from the fine-tuning experiments. Although standardly used for transfer learning, fine-tuning does seem beneficial to remove the dataset bias. Together with the failure of existing adaptive approaches, this questions whether methods effectively used in transfer and domain adaptation settings should be considered automatically as suitable for dataset bias, and vice versa.

How to leverage over the power of deep learning methods to attack the dataset bias problem in all its complexity, well represented by our proposed experimental setup, is open for research in future work.

References

1. Agrawal, P., Girshick, R., Malik, J.: Analyzing the performance of multilayer neural networks for object recognition. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
2. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: NIPS (2007)
3. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: BMVC (2014)
4. Chatfield, K., Simonyan, K., Zisserman, A.: Efficient on-the-fly category retrieval using convnets and gpus. In: ACCV (2014)
5. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)
6. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: ICML (2014), https://github.com/UCB-ICSI-Vision-Group/decaf-release/
7. Duan, L., Tsang, I.W., Xu, D., Chua, T.S.: Domain adaptation from multiple sources via auxiliary classifiers. In: ICML (2009)
8. Everingham, M., Gool, L.V., Williams, C.K., Winn, J., Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. IJCV 88(2) (2010)
9. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009)
10. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Comput. Vis. Image Underst. 106(1), 59–70 (2007)
11. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV (2013)
12. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: ICML (2013)
13. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR (2012)
14. Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: ECCV (2014)
15. Griffin, G., Holub, A., Perona, P.: Caltech 256 object category dataset. Tech. Rep. UCB/CSD-04-1366, California Institue of Technology (2007)
16. Hoffman, J., Guadarrama, S., Tzeng, E., Hu, R., Donahue, J., Girshick, R., Darrell, T., Saenko, K.: LSDA: Large scale detection through adaptation. In: NIPS (2014)
17. Hoffman, J., Kulis, B., Darrell, T., Saenko, K.: Discovering latent domains for multisource domain adaptation. In: ECCV (2012)
18. Hoffman, J., Tzeng, E., Donahue, J., Jia, Y., Saenko, K., Darrell, T.: One-shot adaptation of supervised deep convolutional models (2014)
19. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A., Torralba, A.: Undoing the damage of dataset bias. In: ECCV (2012)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
21. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: ICRA (2011)
22. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between class attribute transfer. In: CVPR (2009)

23. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: CVPR (2003)
24. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV (1999)
25. Microsoft: Microsoft Research Cambridge Object Recognition Image Database. http://research.microsoft.com/en-us/downloads/b94de342-60dc-45d0-830b-9f6eff91b301/default.aspx (2005)
26. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: CVPR (2014)
27. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN Features off-the-shelf: an Astounding Baseline for Recognition. In: arXiv:1403.6382 (2014)
28. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. In: arXiv:1409.0575 (2014)
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) pp. 1–42 (April 2015)
30. S. Chopra, S.B., Gopalan, R.: DLID: Deep learning for domain adaptation by interpolating between domains. In: ICML Workshop on Challenges in Representation Learning, 2013 (2013)
31. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV (2010)
32. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR (2014)
33. Tommasi, T., Tuytelaars, T.: A testbed for cross-dataset analysis. In: ECCV workshop on Task-CV (2014), https://sites.google.com/site/crossdataset/
34. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR (2011)
35. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient object category recognition using classemes. In: ECCV (2010)
36. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR (2010)
37. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014)
38. Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based R-CNNs for fine-grained category detection. In: ECCV (2014)