

Chapter 2

A Deeper Look at Dataset Bias

Tatiana Tommasi, Novi Patricia, Barbara Caputo and Tinne Tuytelaars

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 chapter we propose to verify the potential of the CNN features when facing the dataset bias problem. With this purpose we introduce a large testbed for cross-dataset analysis and we discuss the challenges faced to create two comprehensive experimental setups by aligning twelve existing image databases. We conduct a series of analyses looking at how the 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.

T. Tommasi (✉) · N. Patricia · B. Caputo
University of Rome La Sapienza, Rome, Italy
e-mail: tommasi@dis.uniroma1.it

N. Patricia
e-mail: novi.patricia@idiap.ch

B. Caputo
e-mail: caputo@dis.uniroma1.it

N. Patricia
EPL Lausanne, CH, Lausanne, Switzerland

T. Tuytelaars
KU Leuven, ESAT, PSI, IMEC, Leuven, Belgium
e-mail: Tinne.Tuytelaars@esat.kuleuven.be

2.1 Introduction

Since its spectacular success in the 2012 edition of the Imagenet Large Scale Visual Recognition Challenge (ILSVRC, [404]), deep learning has dramatically changed the research landscape in visual recognition [275]. By training a Convolutional Neural Network (CNN) over millions of data it is possible to get impressively high quality object annotations [5] and detections [579]. A large number of studies have recently proposed improvements over the CNN architecture of Krizhevsky et al. [275] with the aim to better suit an ever increasing typology of visual applications [88, 236, 579]. 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 [384]. Indeed, several works demonstrated that DeCAF (as well as Caffe [128], Overfeat [418], VGG-CNN [66], etc.) can be used as powerful image descriptors [66, 203]. The improvements obtained by 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 work is to contribute to answering this question when focusing on the bias of existing image collections. The *dataset bias* problem was presented and discussed for the first time in [489]. 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, 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 aspects induce a generalization issue 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 [128, 573]. However, the evaluation is generally restricted to controlled cases where the data variability is limited to specific visual domain shift [128, 241] or some images extracted from the testing collection are available at training time [349, 573]. Here we revisit and scale up the analysis, making three contributions:

1. We introduce a large-scale testbed for cross-dataset analysis and discuss the challenges faced when aligning twelve existing image datasets (see Fig. 2.1).
2. We define two comprehensive experimental setups and we assess on them the performance of the CNN features for dataset bias.
3. We propose a new measure to evaluate quantitatively the ability of a given algorithm to address the dataset bias. As opposed to what proposed previously in the literature [489], 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



Fig. 2.1 We show here one image example extracted from each of the 12 datasets (*columns*) for 7 object categories (*rows*): mug, bonsai, fire-hydrant, car, cow, bottle, horse. The empty positions indicate that the corresponding dataset is not annotated for the considered class

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 cannot be applied 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 of new directions, able to accommodate at the same time the potential of deep learning and the difficulties of large-scale cross-database generalization.

Related Work. The existence of several data related issues in any area of automatic classification technology was first discussed by Hand in [220]. The first sign of peril in image collections was indicated in presenting the Caltech256 dataset [215] where the authors recognized the danger of learning ancillary cues of the image collection (e.g. characteristic image size) instead of intrinsic features of the object categories. However, only recently this topic has been really put under the spotlight for computer vision tasks by Torralba and Efros [489]. Their work pointed out the idiosyncrasies of existing image datasets: the evaluation of cross-dataset performance revealed that standard detection and classification methods fail because the uniformity of training and test data is not guaranteed.

This initial analysis of the *dataset bias* problem gave rise to a series of works focusing on how to overcome the specific image collection differences and learn robust classifiers with good generalization properties. The proposed methods have been mainly tested on binary tasks (object versus rest) where the attention is focused on categories like *car* or *person* which are common among six popular datasets: SUN, LabelMe, PascalVOC, Caltech101, Imagenet, and MSRC [489]. A further group of three classes was soon added to the original set (*bird*, *chair* and *dog*) defining a total of five object categories over the first four datasets listed before [151, 268]. A larger scale analysis in terms of categories was proposed in [389] by focusing on 84 classes of Imagenet and SUN, while a study on how to use weakly labeled Bing images to classify Caltech256 samples was proposed in [36]. Finally the problem of partially overlapping label sets among different datasets was considered in [485].

Together with the growing awareness about the characteristic signature of each existing image set, the related problem of *domain shift* has also emerged. Given a source and target image set with different marginal probability distributions, any learning method trained on the first will present lower performance on the second. In real life settings it is often impossible to have full control on how the test images will differ from the original training data and an adaptation procedure to remove the domain shift is necessary. An efficient (and possibly unsupervised) solution is to learn a shared representation that eliminates the original distribution mismatch. Different methods based on subspace data embedding [164, 200], metric [407, 483] and vocabulary learning [377] have been presented. As already mentioned above, several works have also demonstrated that deep learning architectures may produce domain invariant descriptors through highly nonlinear transformation of the original features [128]. Domain adaptation (DA) algorithms have been mostly evaluated on the Office (OFF31) dataset [407] and Office-Caltech (OC10) [200] containing office-related object categories from three respectively 4 domains.

Despite their close relation, visual domain and dataset bias are not the same. Domain adaptation solutions have been used to tackle the dataset bias problem, but domain discovery approaches have shown that a single dataset may contain several domains [238] while a single domain may be shared across several datasets [197]. Moreover, the domain shift problem is generally considered under the covariate shift assumption with a fixed set of classes shared by the domains and analogous conditional distributions. On the other hand, different image datasets may contain different object classes.

Here we make up the lack of a standard testbed for large-scale cross-dataset analysis and we evaluate the effect of the CNN features for this task. We believe that widening the attention from few shared classes to the whole dataset structures can reveal much about the nature of the biases and on the effectiveness of the proposed representations and algorithmic solutions.

2.2 A Large-Scale Cross-Dataset Testbed

In this section we first give a brief description of the considered image datasets, created and used before for object categorization:

ETH80 [293] was created to facilitate the transition from object identification (recognize a specific object instance) to categorization (assign the correct class label to an object instance never seen before). It contains 8 categories and 10 toy objects each represented by 41 images captured against a blue background from viewpoints spaced equally over the upper viewing hemisphere.

Caltech101 [160] contains 101 object categories and was the first large-scale collection proposed as a testbed for object recognition algorithms. Each category contains a different number of samples between 31 and 800 images. The images have little or no clutter with the objects centered and presented in a stereotypical pose.

Caltech256 [215]. Differently from the previous case the images in this dataset were not manually aligned, thus the objects appear in several different poses. This collection contains 256 categories with between 80 and 827 images per class.

Bing [36] contains images downloaded from the Internet for the same set of 256 object categories of the previous collection. Text queries give as output several noisy images which are not removed, resulting in a weakly labeled collection. The number of samples per class goes from a minimum of 197 to a maximum of 593.

Animals with Attributes (AwA) [287] presents a total of 30475 images of 50 animal categories. Each class is associated to an 85-element vector of numeric attribute values that indicate general characteristics shared between different classes. The animals appear in different pose and at different scales in the images.

a-Yahoo [157]. As the previous one, this dataset was collected to explore attribute descriptions. It contains 12 object categories with a minimum of 48 and a maximum of 366 samples per class.

MSRCORID [331]. The Microsoft Research Cambridge Object Recognition Image Database contains a set of digital photographs grouped into 22 categories spanning over objects (19 classes) and scenes (3 classes).

PascalVOC2007 [148]. The Pascal Visual Object Classes dataset contain 20 object categories and a total of 9963 images. Each image depicts objects in realistic scenes and may contain instances of more than one category. This dataset was used as testbed for the Pascal object recognition and detection challenges in 2007.

SUN [541] contains a total of 142165 pictures¹ and it was created as a comprehensive collection of annotated images covering a large variety of environmental scenes, places and objects. Here the objects appear at different scales and positions in the

¹Version available on December 2013 at http://labelme.csail.mit.edu/Release3.0/Images/users/antonio/static_sun_database/ and the list of objects reported at <http://groups.csail.mit.edu/vision/SUN/>.

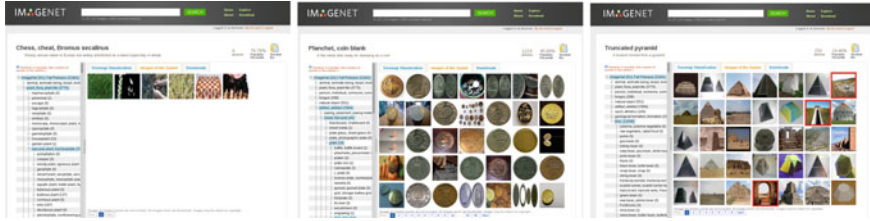


Fig. 2.2 Three cases of Imagenet categories. *Left* some images in class *chess* are wrongly labeled. *Middle*: the class *planchet* or coin blank contains images that can be more easily labeled as *coin*. *Right* the images highlighted with a *red square* in the class *truncated pyramid* do not contain a pyramid (best viewed in color and with magnification)

images and many of the instances are partially occluded making object recognition and categorization very challenging.

Office [407]. This dataset contains images of 31 object classes over three domains: the images are either obtained from the Amazon website, or acquired with a high-resolution digital camera (DSLR), or taken with a low resolution webcam. The collection contains a total of 4110 images with a minimum of 7 and a maximum of 100 samples per domain and category.

RGB-D [284] is similar in spirit to ETH80 but it was collected with a Kinect camera, thus each RGB image is associated to a depth map. It contains images of 300 objects acquired under multiple views and organized into 51 categories.

Imagenet [120]. At the moment this collection contains around 21800 object classes organized according to the Wordnet hierarchy.

2.2.1 Merging Challenges

There are two main challenges that must be faced when organizing and using at once all the data collections listed before. One is related to the alignment of the object classes and the other is the need for a shared feature representation.

Composing the datasets in a single corpus turned out to be quite difficult. Even if each image is labeled with an object category name, the class alignment is tricky due to the use of different words to indicate the very same object, for instance *bike* versus *bicycle* and *mobilephone* versus *cellphone*. Sometimes the different nuance of meaning of each word is not respected: *cup* and *mug* should indicate two different objects, but the images are often mixed; *people* is the plural of *person*, but images of this last class often contain more than one subject. Moreover, the choice of different ontology hierarchical levels (*dog* versus *dalmatian* versus *greyhound*, *bottle* versus *water-bottle* versus *wine-bottle*) complicates the combination. Psychological studies demonstrated that humans prefer entry-level categories when naming visual objects [350], thus when combining the datasets we chose “natural” labels that correspond to

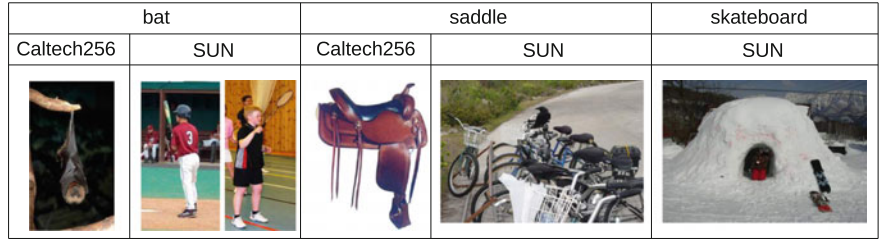


Fig. 2.3 Three categories with labeling issues. The class *bat* has different meanings both across datasets and within a dataset. A *saddle* can be a seat to ride a horse or a part of a bicycle. A *skateboard* and a *snowboard* may be visually similar, but they are not the same object

intermediate nodes in the Wordnet hierarchy. For instance, we used *bird* to associate humming bird, pigeon, ibis, flamingo, rooster, cormorant, ostrich and owl, while *boat* covers kayak, ketch, schooner, speed boat, canoe and ferry. In the cases in which we combine only two classes we keep both their names, e.g. *cup* and *mug*.

In the alignment process we came across a few peculiar cases. Figure 2.2 shows samples of three classes in Imagenet. The category *chess board* does not exist at the moment, but there are three classes related to the word *chess*: chess master, chessman or chess piece, chess or cheat or bromus secalinus (we use “or” here to indicate different labels associated to the same synset). This last category contains only a few images but some of them are not correctly annotated. The categories *coin* and *pyramid* are still not present in Imagenet. For the first, the most closely related class is *planchet* or *coin blank*, which contains many examples of what would be commonly named as a coin. For the second, the most similar *truncated pyramid* contains images of some non-truncated pyramids as well as images not containing any pyramids at all. In general, it is important to keep in mind that several of the Imagenet pictures are weakly labeled, thus they cannot be considered as much more reliable than the corresponding Bing images. Imagenet users are asked to clean and refine the data collection by indicating whether an image is a typical or wrong example.

We noticed that the word *bat* usually indicates the flying mammal except in SUN where it refers to the baseball and badminton bat. A *saddle* in Caltech256 is the supportive structure for a horse rider, while in SUN it is a bicycle seat. Tennis shoes and sneakers are two synonyms associated to the same synset in Imagenet, while they correspond to two different classes in Caltech256. In SUN, there are two objects annotated as skateboards, but they are in fact two snowboards. Some examples are shown in Fig. 2.3. We disregarded all these ambiguous cases and we do not consider them in the final combined setups.

Although many descriptors have been extracted and evaluated separately on each image collection, the considered features usually differ across datasets. Public repositories with pre-calculated features exist for Caltech101 and Caltech256, Bing and

Caltech256, and for a set of five classes out of four datasets.² Here we consider the group of twelve datasets listed in the previous section and extracted the same features from all of them defining a homogeneous reference representation for cross-dataset analysis.

2.2.2 Data Setups and Feature Descriptor

Dense set. Among the considered datasets, the ones with the highest number of categories are Caltech256, Bing, SUN and Imagenet. In fact the last two are open collections progressively growing in time. Overall they share 114 categories: some of the 256 object categories are missing at the moment in Imagenet but they are present in SUN (e.g. desk-globe, fire-hydrant) and vice versa (e.g. butterfly, pram). Out of this shared group, 40 classes (see Fig. 2.4) contain more than 20 images per dataset and we selected them to define a dense cross-dataset setup. We remark that each image in SUN is annotated with the list of objects visible in the depicted scene: we consider an image as a sample of a category if the category name is in the mentioned list.

Sparse set. A second setup is obtained by searching over all the datasets for the categories which are shared at least by four collections and that contain a minimum of 20 samples. We allow a lower number of samples only for the classes shared by more than four datasets (i.e. from the fifth dataset on the images per category may be less than 20). These conditions are satisfied by 105 object categories in Imagenet overlapping with 95 categories of Caltech256 and Bing, 89 categories of SUN, 34 categories of Caltech101, 17 categories of Office, 18 categories of RGB-D, 16 categories of AwA and PascalVOC07, 13 categories of MSRCORID, 7 categories of ETH80 and 4 categories of a-Yahoo. The histogram in Fig. 2.5 shows the defined sparse set and the number of images per class: the category *cup and mug* is shared across nine datasets, making it the most popular one.

Representation. We release the cross-dataset with three feature representations:

- **BOWsift:** dense SIFTs have been among the most widely used handcrafted features in several computer vision tasks before the advent of the CNN representations, thus we decided to use this descriptor as reference and we adopted the same extraction protocol proposed in the Imagenet development kit³ by running their code over the twelve considered datasets. Each image is resized to have a max size length of no more than 300 pixels and SIFT descriptors [315] are computed on 20×20 overlapping patches with a spacing of 10 pixels. Images are further downsized (to 1/2 and 1/4 of the side length) and more descriptors are computed. We used the visual vocab-

²Available respectively at <http://files.is.tue.mpg.de/pgehler/projects/iccv09/>, <http://vlg.cs.dartmouth.edu/projects/domainadapt/>, <http://undoingbias.csail.mit.edu/>.

³www.image-net.org/download-features.

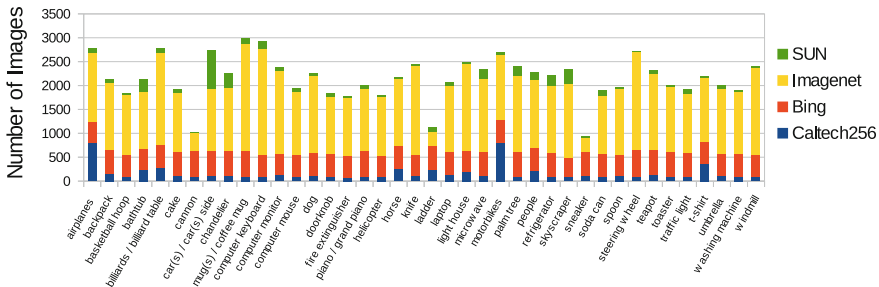


Fig. 2.4 Stack histogram showing the number of images per class of our cross-dataset dense setup

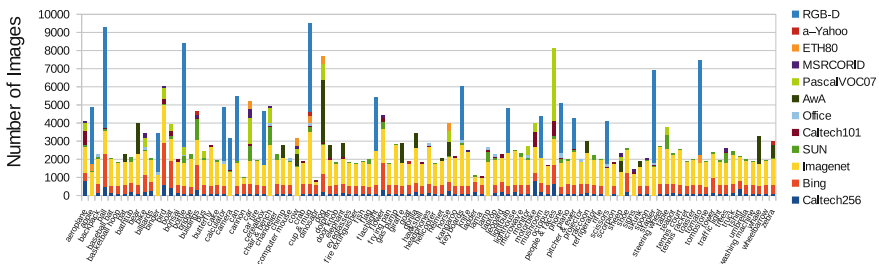


Fig. 2.5 Stack histogram showing the number of images per class of our cross-dataset sparse setup (best viewed in color and with magnification)

ulary of 1000 words provided with the mentioned kit to obtain the Bag of Words (BOW) representation [108, 445]: it was built over the images of the 1000 classes of the ILSVRC2010 challenge [404] by clustering a random subset of 10 million SIFT vectors.

• **DeCAF6, DeCAF7:** the mean-centered raw RGB pixel intensity values of all the collection images (warped to 256×256) are given as input to the CNN architecture of Krizhevsky et al. [275] by using the DeCAF implementation.⁴ The activation values of the 4096 neurons in the 6-th and 7-th layers of the network are considered as image descriptors [128].

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 DA methods. We mostly focus on the results obtained with the DeCAF features and use the BOWsift representation as a reference baseline.

Evaluation Protocol. Our basic experimental setup considers both in-dataset and cross-dataset evaluations. With *in-dataset* we mean training and testing on samples extracted from the same dataset, while with *cross-dataset* we indicate experiments where training and testing samples belong to different collections. We use *Self* to

⁴<https://github.com/UCB-ICSI-Vision-Group/decaf-release/>.

specify the in-dataset performance and *Mean Other* for the average cross-dataset performance over multiple test collections.

One way to quantitatively evaluate the cross dataset generalization was previously proposed in [489]. It consists of measuring 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 $CD = (1 + \exp^{-((Self - Mean\ Others)/100)})^{-1}$. It 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.

2.3 Studying the Sparse Set

Dataset Recognition. One of the effects of the capture bias is that it makes any dataset easily recognizable. We want to evaluate whether this effect is enhanced or decreased by the use of the CNN features. To do it we run the *name the dataset* test [489] 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 Fig. 2.6. 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 a large number of training samples. From the confusion matrices (middle and right in Fig. 2.6) 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 to each other, nor how a specific class model will generalize to other datasets. We look into this problem in the following paragraph.

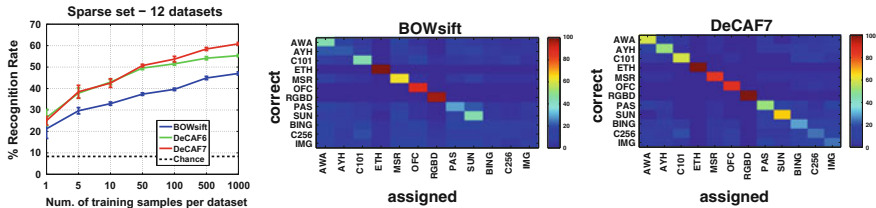


Fig. 2.6 Name the dataset experiment over the sparse setup with 12 datasets. The title of each confusion matrix indicates the feature used for the corresponding experiments

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 of the sparse set contain images labeled with these object classes: PascalVOC07 (P), SUN (S), ETH80 (E), and MSCORID (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 (limited by the number of cow images in SUN). We repeat the sample selection 10 times and the average precision results obtained by linear SVM classification are presented in the matrices of Table 2.1.

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 from 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 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 [128] which cover several vehicles and animal categories. The class *car* is in this set, but *cow* is not. This intrinsically induces 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

Table 2.1 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. P,S,E,M stand respectively for the datasets Pascal VOC07, SUN, ETH80, MSCORID

	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

cross-dataset results indicates that the negative set bias maintains 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 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 [241], 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 [268] to overcome the dataset bias and 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.

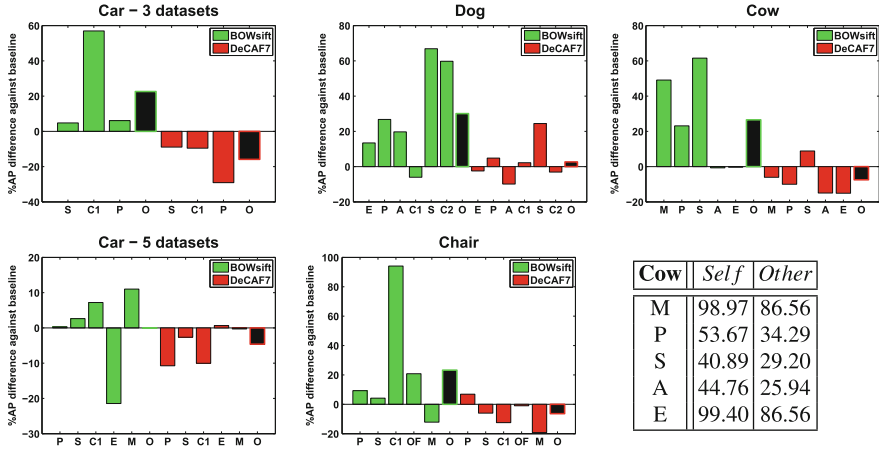


Fig. 2.7 Percentage difference in average precision between the results of *Unbias* and the baseline *All* over each target dataset. P,S,E,M,A,C1,C2,OF stand respectively for the datasets Pascal VOC07, SUN, ETH80, MSRCORID, AwA, Caltech101, Caltech256 and Office. O indicates the overall value, i.e. the average of the percentage difference over all considered datasets (*shown in black*)

We run the experiments focusing on the classes *car*, *cow*, *dog* and *chair* and reproducing a similar setup to what previously used in [268]. 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 the round for testing while all the others are used as sources of training samples.

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 Fig. 2.7. 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 decrease, 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 Fig. 2.7 we present the results obtained with the class *cow* together with the average precision per dataset when using DeCAF7. Specifically, 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 trained 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.

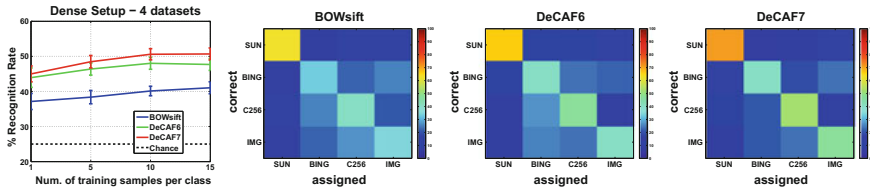


Fig. 2.8 Name the dataset experiment over the dense setup with 4 datasets. The title of each confusion matrix indicates the feature used for the corresponding experiments

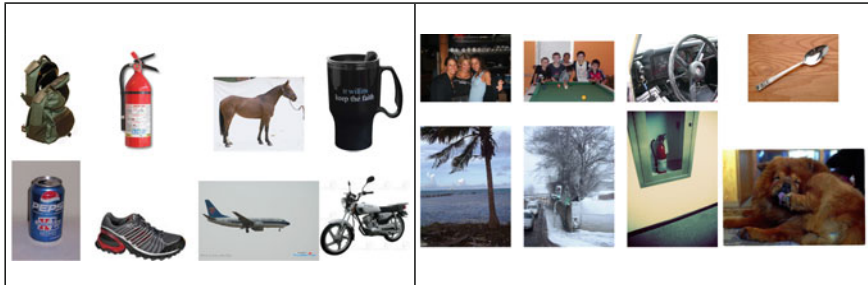


Fig. 2.9 Left Imagenet images annotated as Caltech256 data with BOWsift but correctly recognized with decaf7. Right Caltech256 images annotated as Imagenet by BOWsift but correctly recognized with DeCAF7

2.4 Studying the Dense Set

Dataset Recognition. The 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 Fig. 2.8 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. We can get a more concrete idea of the DeCAF performance by looking at Fig. 2.9. Here images on the left present Imagenet images that have been assigned to Caltech256 with BOWsift but which are correctly recognized with DeCAF7. Images on the right contain instead Caltech256 images wrongly annotated as Imagenet samples by BOWsift but correctly labeled with DeCAF7. Considering the white background and standard pose that characterize Caltech256 images, together with the less stereotypical content of Imagenet data, the mistakes of BOWsift can be visually justified, nevertheless the DeCAF features overcome them.

Table 2.2 Multi-class 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					
train	C256	25.15	15.05	9.35	51.5	0.53		C256	73.15	56.05	20.20	47.9	0.58							
	IMG	14.50	17.85	9.05					34.0	0.52	train				IMG	64.10	64.90	22.65	33.2	0.55
	SUN	7.70	8.00	13.55					42.1	0.51					SUN	21.35	23.15	30.05	25.9	0.52
		C256	IMG test	SUN				C256	IMG test	SUN										

Since all the datasets contain the same object classes, we are in fact reproducing a setup generally adopted for DA [164, 200]. By identifying each dataset with a domain, we can interpret the results of this experiment as an indication of the domain divergence [31] 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 multi-class SVM per dataset. Due to its noisy nature we exclude Bing here and we dedicate more attention to it in the next paragraph.

The average recognition rate results over 10 data splits are reported in Table 2.2. 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.

To visualize the effect of the dataset-bias per class we present the separate recognition rate in Fig. 2.10. Specifically we consider the case of training on Caltech256. From the top plot we can see that *motorcycle*, *aeroplane* and *car* are the objects better recognized when testing on Caltech256 with BOWsift and they are also the classes that mostly contribute to the recognition drop when testing on ImageNet and SUN. On the other hand, the classes *steering-wheel*, *windmill*, *bathtub*, *lighthouse* and

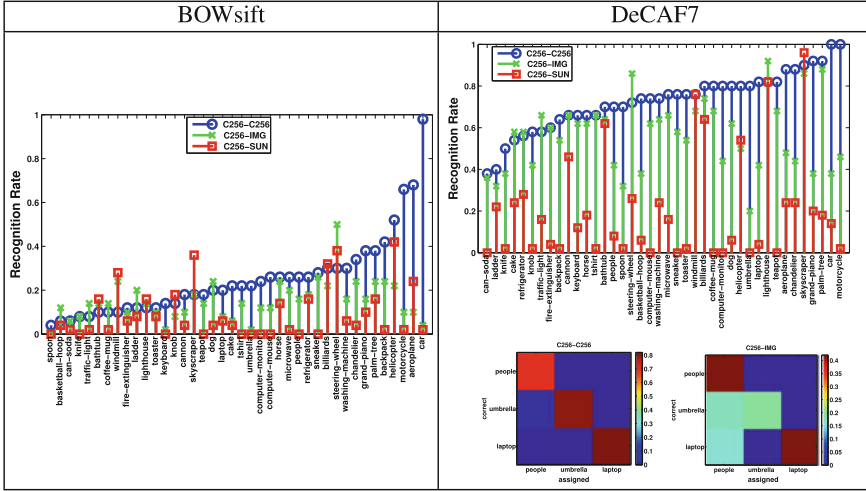


Fig. 2.10 Recognition rate per class from the multi-class cross-dataset generalization test. C256, IMG and SUN stand respectively for Caltech256, ImageNet and SUN datasets. We indicate with “train-test” the pair of datasets used in training and testing

skyscraper are better recognized on SUN and/or Imagenet than on Caltech256. All these last objects occupy most part of the image in all the collections and present less dataset-specific characteristics. When looking at the results with DeCAF7, *motorcycle* and *car* are still among the classes with the highest cross-dataset recognition difference, together with *people*, *spoon*, *umbrella*, *basketball-hoop* and *laptop*.

As already indicated by the binary experiments, even these results confirm that the dataset bias is in fact class dependent and that using DeCAF does not automatically solve the problem. A further remark can be done here about Imagenet. Although often considered as one of the less biased collections it actually presents a specific characteristic: the images are annotated with a single label but in fact may contain more than one visual category. In particular, its images often depict people even when they are labeled with a different class name. As a demonstration we report at the bottom of Fig. 2.10 a sub-part of the confusion matrix when training on Caltech256 and testing both on itself and on Imagenet. The results show that people are recognized in the class umbrella and laptop with relevant influence on the overall annotation errors.

Noisy Source Data and Domain Adaptation. Until now we have discussed and demonstrated empirically that the difference among two data collections can actually 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 [66]. Some DA strategies seem perfectly suited for this task (see Fig. 2.11 top part) and we use them here to evaluate the cross-dataset generalization performance when training on Bing (noisy

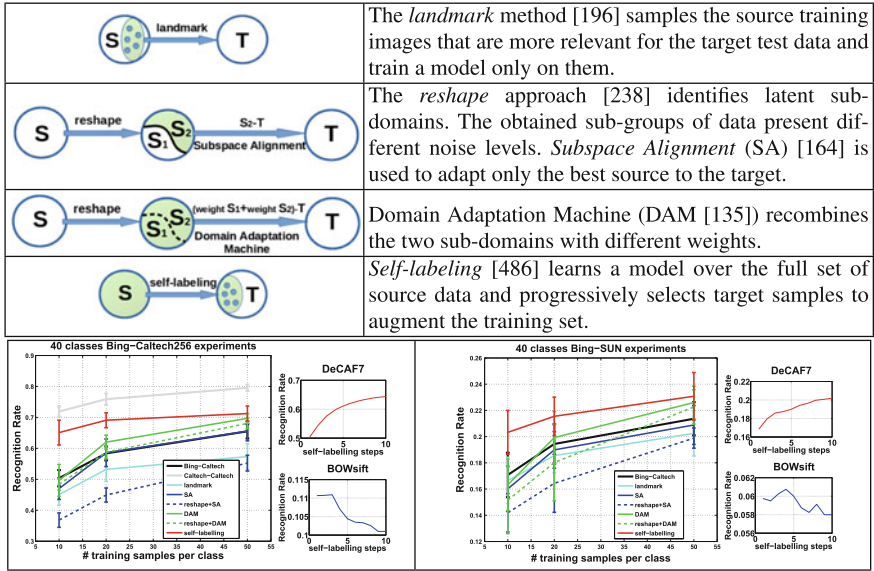


Fig. 2.11 *Top* schematic description of the used DA methods. *Bottom* Results of the Bing-Caltech256 and Bing-SUN experiments with DeCAF7. We report the performance of different DA methods (big plots) together with the recognition rate obtained in 10 subsequent steps of the self-labeling procedure (small plots). For the last ones we show the performance obtained both with DeCAF7 and with BOWsift when having originally 10 samples per class from Bing

object-centric source domain) and testing on Caltech256 and SUN (respectively an object-centric and a scene-centric target domain). We consider an increasing number of training images per class from 10 to 50 and we test on 30 images per class on Caltech256 and 20 images per class on SUN. The experiments are repeated for 10 random data splits.

The obtained results are shown in Fig. 2.11—bottom part, go in the same direction of what 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 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* [486] consistently improves the original results with a significant gain in performance especially when only a reduced set of training images per class is available. 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: this can be appreciated by looking at the recognition rate obtained over ten iterations of the target selection procedure, considering in particular the comparison against the corresponding performance obtained when using BOWsift (see the small plots in Fig. 2.11).

Fine-Tuning. As indicated in Sect. 2.2.2 the CNN features used for our analysis were obtained from pre-trained network whose parameters remain untouched. Previous work showed that modifying the network by fine-tuning before using it for recognition on a new task can be useful [349, 573]. We clarify here that this standard fine-tuning process does not fit in the dataset bias setting used for our study.

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) is (D, D', D') . In our analysis we have instead a different condition: (D, D', D'') where D' consists in a reduced amount of labeled data, while D'' is the test set extracted from a collection different from D' . It has been demonstrated that fine-tuning on a small amount of samples provides bad results [241] and it makes the features dataset-specific [66], which can only increase the bias. By using the Caffe CNN implementation we fine-tuned the Imagenet pre-trained network on the dense set, specifically on Caltech256 (5046 train images, 40 classes) and SUN (3015 train images, 40 classes), reserving respectively 1500 and 1300 images from these two datasets as test data. The in-dataset and cross-dataset experimental results are: $(\text{Imagenet}, \text{Caltech256}, \text{Caltech256}) = 86.4\%$; $(\text{Imagenet}, \text{SUN}, \text{SUN}) = 41.1\%$. $(\text{Imagenet}, \text{SUN}, \text{Caltech256}) = 37.5\%$; $(\text{Imagenet}, \text{Caltech256}, \text{SUN}) = 25.7\%$. Compared with what presented in Table 2.2 these results show the advantage of the fine-tuning in terms of overall recognition rate. However they also confirm that the fine-tuning process does not remove the bias ($86.4 > 25.7\%$; $41.1 > 37.5\%$) and that using the wrong dataset to refine the network can be detrimental ($86.4 > 37.5\%$; $41.1 > 25.7\%$).

2.5 Conclusion

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 proposed a large-scale cross-dataset testbed defined over 12 existing datasets organized into two setups, and we focused on DeCAF features 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 adds 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. This questions whether methods effectively used in DA should be considered automatically as suitable for dataset bias, and vice versa.

How to leverage over the power of deep learning methods to attack this problem in all its complexity, well represented by our proposed experimental setup, is open for research in future work. We consider this work as the first step of a wider project (the official webpage <https://sites.google.com/site/crossdataset/>): we already calculated and released new versions of the CNN features obtained with different architectures and by using different pre-training datasets on which we are planning an even larger experimental evaluation.

Domain Adaptation in Computer Vision Applications

Csurka, G. (Ed.)

2017, X, 344 p. 107 illus., 101 illus. in color., Hardcover

ISBN: 978-3-319-58346-4