Predicting an Object Location using a Global Image Representation

Jose A. Rodriguez-Serrano and Diane Larlus
Computer Vision Group, Xerox Research Centre Europe
(jose-antonio.rodriguez,diane.larlus)@xrce.xerox.com

Abstract

We tackle the detection of prominent objects in images as a retrieval task: given a global image descriptor, we find the most similar images in an annotated dataset, and transfer the object bounding boxes. We refer to this approach as data driven detection *(DDD), that is an alternative to sliding windows. Previous works have used similar notions but with task-independent similarities and representations,* i.e. *they were not tailored to the end-goal of localization. This article proposes two contributions: (i) a metric learning algorithm and (ii) a representation of images as object probability maps, that are both optimized for detection. We show experimentally that these two contributions are crucial to DDD, do not require costly additional operations, and in some cases yield comparable or better results than state-of-the-art detectors despite conceptual simplicity and increased speed. As an application of prominent object detection, we improve fine-grained categorization by pre-cropping images with the proposed approach.*

1. Introduction

This paper deals with the problem of prominent object detection, where the goal is to predict the region containing the relevant subject (or object of interest) in an image, as opposed to other regions containing background or non-relevant objects. This problem is encountered in a variety of computer vision applications. For example, in image thumbnailing [23] or auto-cropping the goal is to detect the boundaries of the foreground object. In mobile phone applications, such as product search [28] or *leafsnap.com* [16], users take pictures of an "object of interest", and localizing the object is often necessary for the subsequent processing steps. Another scenario where this problem is found is fine-grained categorization [8, 29, 36, 35]. In fine-grained categorization an image contains an object of a parent class (e.g. bird, dog, car), and the goal is to classify it into one of the more specific sub-classes (*e.g.* dog breed, bird species, car makes and models). The localization of the prominent object is a key cue that can be used to improve this difficult recognition task (as shown in section 6).

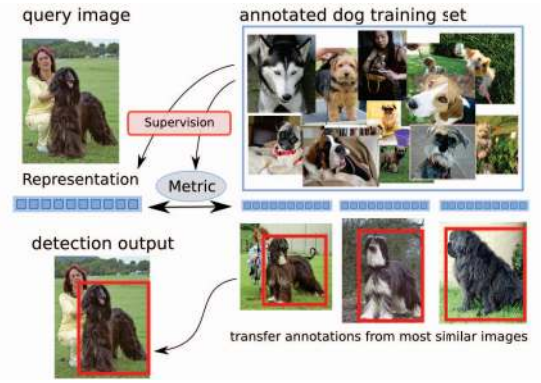

Figure 1. We aim at detecting the prominent object of an image using a global image representation. A query image is compared to an annotated set, and the nearest neighbors transfer their annotations. For this retrieval process, we use supervision to learn an image representation and a metric geared toward detection.

As these examples show, the definition of *prominent* or *relevant* object is *application-dependent*, and we assume this is defined by a training set with annotated bounding boxes. This can be considered as a special case of detection, with a single object per image. Research in detection has converged to combining a template descriptor (*e.g.* HOG [6] or its extension to deformable parts [10]) with sliding windows. While these methods have obtained impressive results in multi-object, multi-class scenarios (such as PASCAL VOC [9]), one drawback is that they require to classify millions of windows per image. Although methods for accelerating sliding window search have been proposed, a large number of window descriptors have to be extracted and classified independently.

In this paper, we aim at an efficient solution, and propose to extract a single global feature for the input image and to estimate the prominent object location directly from this feature vector, avoiding sliding windows. In this context, we note that if the descriptor contains spatially-variant information (which can be achieved by spatial pooling), then the similarity between those global descriptors provides a

strong cue for object location: neighbors tend to coincide not only in appearance but also in location (see Fig 1).

This suggests a simple retrieval-based method for prominent object localization: given an input image, find the nearest images from a database (using the global descriptor), and transfer the bounding box of the most similar image. This data-driven detection[1] (DDD) approach has certain advantages. First, detection is performed at the ease and efficiency of a retrieval operation. Second, it allows handling any object shape and does not rely on the rectangular and rigid object assumption of the sliding-window approaches. Finally, as detection is obtained using a global image descriptor, it intrinsically leverages context for detection.

However, this idea presents some initial issues. First, having images of the object at any possible location in the training set seems infeasible – although this effect can be reduced by combining the top K results. But, more importantly, previous literature suggests negative evidence against that approach. Several works [27, 23, 26, 30, 20] have exploited this "global image transfer", but in all cases they needed to combine it with more complex and sometimes costly refinements, which could indicate that the retrieval step is not sufficient on its own.

We believe that a key limitation of the previous idea is that it uses generic image representations and similarities that we call "task-independent", and that are disconnected from the end-task of detection. This paper proposes two improvements:

A task-aware similarity function. We aim at predicting that two images have similar localization annotations directly from their image features. We apply metric learning to enforce image pairs with high overlap between detection rectangles to be more similar than images with no (or small) overlap.

A task-aware representation. We propose to use a compact image representation that is constructed from the probability of each image patch to belong to the object. This requires some knowledge about the object to locate, but allows representing the image as a probability map of the object location. Since there exists a strong correlation between the true detection and such a probability map, these features are well-suited to data-driven detection.

Both contributions use supervision to connect the features and similarity to the detection task, by converting an initial assumption (similar images tend to have similar layouts) into an actual optimization step (we *learn* what makes two layouts similar). Experiments indicate that these two components significantly improve data-driven detection. Also, more compact representations are obtained that lead to a faster retrieval at test time. Despite their simplicity at test time, these two components lead to an accuracy that on some datasets is on par with the deformable part model (DPM) [10], which is state-of-the-art for detection. Finally, we show in fine-grained categorization experiments that we improve classification accuracy by concentrating on the region predicted by a DDD.

The paper is organized as follows. § 2 reviews previous work. § 3 summarizes the general principle of data-driven detection. § 4 and 5 present the proposed task-aware similarity and representation, respectively. § 6 presents experiments on three detection tasks. § 7 concludes the paper.

2. Related Work

Detection. De-facto standard detection methods [6, 10] combine a template representation and a sliding window approach. Among them, HOG descriptors [6] have shown big success for fully rigid objects in multi-class and multi-object settings. DPM [10] builds on top of HOG, combining it with deformable parts to be robust to small object deformations. Both methods still have issues with flexible objects [9]. Additional abundant work on detection has been published but the most successful methods cast detection as a classification problem: a large number of classification operations are performed (each possible sub-window is classified as containing the object or not). In contrast, in this paper we would like to take a data-driven approach, and cast detection as a retrieval problem.

The class-generic objectness measure [2] has been used for preprocessing, *e.g.* to improve or speed up detection [15]. It differs from DDD in spirit (it is not application dependent) and in practice (it is a sliding window approach).

Data-Driven Localization. The concept of transferring bounding boxes (or pixel-level masks) from the nearest neighbors of an image has been successfully exploited in previous work. Two main strategies exist: transfer at sub-window level and transfer at full-image level.

In the first strategy, approaches still perform sliding window search and each sub-window is used as a query for which nearest-neighbor similarity is computed [31, 22]. A similar case is the figure-ground segmentation of [15] where sliding window search is replaced by an objectness detector [2]. Although these works clearly have a data-driven component that is key to their success (the scoring of sub-windows), they still compare all sub-windows of an image against a database, and pay a complexity price that it at least as big as the one paid by sliding window approaches.

The second strategy performs transfer at full-image level and exploits the intuition that similar images tend to have similar segmentations [20, 23, 26, 30] or detections [27]. All these works boil-down to the same principle: find the nearest neighbors of the input image; and feed the annotations of those to a more complex method. For instance, in

[1] The term *data-driven* is recently used to refer to approaches which successfully reduce complex regression problems (*e.g.* image annotation, geolocalization) to nearest-neighbor transfer in huge datasets [31, 21, 13], exploiting the phenomenon of the *unreasonable effectiveness of data* [12].

[20, 26, 30] a Markov random field is instantiated from the transferred segmentations. Similarly, [27] uses the neighbors to induce priors over object classes and bounding box locations in a graphical model.

We observe that for the second strategy, transferring the labels of the neighbors is crucial to guide the algorithm, but as the retrieval step is based on *task-independent* features and similarities, this is not sufficient. Their results may be due to the more complex models used after the data-driven step. In [15] a global neighbor transfer baseline produces poor results in a multi-object binary segmentation task. Our method belongs to the second strategy, except that we use supervision in the retrieval step.

3. Data-driven detection baseline

Our goal is to infer the bounding box of the prominent object from the global feature vector of the image, using a training set with annotated bounding boxes. Based on the intuition from recent data-driven segmentation works [27, 26, 30, 20, 15], data-driven detection (DDD) can be formalized as follows. We denote by $\{x_i, R_i\}$ a training set of feature-annotation pairs, where $x_i \in \mathbb{R}^D$ denotes the feature vector of the ith image and $R_i \in \mathbb{R}^4$ the ground-truth rectangle indicating the extent of an object of interest in the image (top-left and bottom-right corners). We assume a similarity function between features $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. For a query feature q, $k_n = k(q, x_n)$ denotes its similarity to the nth element of the training set, and $\pi(l)$ the index of the element that ends up in position l after sorting the k_n values in descending order. We seek a function that predicts the rectangle $R(q)$, using the L top-ranked samples: $R(q) = \mathcal{F}(x_{\pi(1)}, \dots x_{\pi(L)})$. A straightforward choice for $\mathcal{F}$ is a non-parametric regression:

$$R(q) = \sum_{l=1}^{L} w_{\pi(l)} R_{\pi(l)}, \tag{1}$$

which expresses the predicted rectangle as a weighted combination of the ground-truth rectangles of the L best ranked samples. We chose $w_i = k_i / \sum_r k_r$. The process is illustrated in Fig. 1.

Proposed baseline. Although the theoretical formulation is valid for any feature and similarity, note that data-driven methods need to be computationally efficient. We therefore prefer using similarities of the form of a Mercer kernel which have known approximate explicit embeddings in finite-dimensional spaces. In this case similarities can be expressed as dot products $k(q, x) = q^T x$, which can be efficiently computed, and the features q and x already encode the explicit embeddings. In this article, we use Fisher Vectors (FV) as our generic representation, as they fulfill the above property and have obtained state-of-the-art results in image categorization [25] and retrieval [24] tasks.

4. Task-aware metric

Our first contribution is a similarity learning algorithm that optimizes a detection criterion. Metric learning[2] approaches [3, 7, 34] aim at obtaining distances or similarities optimized for tasks such as k-NN classification [7, 34] or ranking [3]. All these works make use of categorical labels for samples or pairs. We are not aware of previous works applying metric learning on object location labels. A fundamental difference is that here the label space is continuous and multi-dimensional, not categorical.

Definitions. We assume that a similarity function over annotations $\Psi : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}$ is defined. In the following, if m and n index the images of a labeled data set, we may use the shorthand $\Psi_{mn} = \Psi(R_m, R_n)$.

For object localization, a common similarity is the *overlap score*, defined as the union-to-intersection area ratio (*e.g.* see PASCAL detection challenge [9]):

$$\Psi(R, R') = \frac{Area(R \cap R')}{Area(R \cup R')}. \tag{2}$$

Similarity function. Our goal is to learn a similarity function $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ which ranks images as similarly as possible to the ranking induced by Ψ. Intuitively, this means that image pairs with similar annotations are forced to have similar representations according to the learnt metric.

More precisely, we consider a similarity function which augments the dot product as:

$$k_W(q, x) = q^T W x. \tag{3}$$

Loss function. We define an objective function that quantifies the "loss" of choosing k_W given Ψ on a training set:

$$\mathcal{L}(W) = \sum_{\substack{\forall i,j,k \\ s.t. \Psi(R_i,R_j) > \Psi(R_i,R_k)}} \Delta_{ijk} I[\![k_W(x_i, x_k) > k_W(x_i, x_j)]\!], \tag{4}$$

where $I[\![a]\!]$ equals to 1 if a is true and to 0 if false. In words, for a triplet (i, j, k) ordered such that $\Psi(R_i, R_j) > \Psi(R_i, R_k)$, we check whether $k_W(\cdot, \cdot)$ respects the ordering; if not a cost of Δ_{ijk} is paid (defined later), and we accumulate the costs of all the triplets in the set.

Thus the goal is to minimize Eq. (4) w.r.t. W, but as it is intractable we use the following convex upper-bound:

$$\mathcal{L}(W) = \sum_{\substack{\forall i,j,k \\ s.t. \Psi(R_i,R_j) > \Psi(R_i,R_k)}} \max(0, \Delta_{ijk} + k_W(x_i, x_k) - k_W(x_i, x_j)). \tag{5}$$

Note that Eq. (5) is reminiscent of a margin-rescale hinge loss, commonly employed in structured learning. Following

[2] abusing the language we sometimes use the term "metric learning" as a synonym of "distance learning" or even "similarity learning"

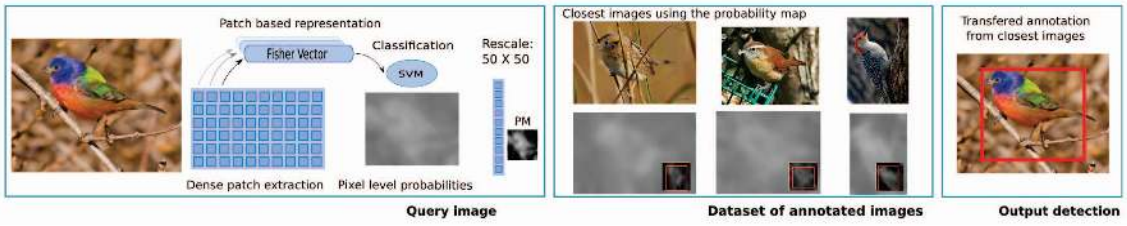

Figure 2. Task-aware representation: patch-level object classifiers are used to represent query images by probability maps. Annotations of training images with similar probability maps are transferred to solve detection.

this analogy, we refer to Δ_{ijk} as the Δ-loss. In practice, the Δ-loss acts as a margin.

The two components of $\mathcal{L}(W)$ that are crucial to deal with the structured output nature of our labels are (i) the Δ-loss defined over rectangles and (ii) the sampling strategy (triplets s.t. $\Psi(R_i, R_j) > \Psi(R_i, R_k)$). We consider three possible choices of Δ_{ijk} : (a) *constant* Δ*-loss*: $\Delta_{ijk} = 1$, (b) *variable* Δ*-loss*: $\Delta_{ijk} = \Psi_{ij}(\Psi_{ij} - \Psi_{ik})$, which encourages that simple cases in the detection space get well-separated in the image space (or pay a big cost if not), (c) *biased sampling*: $\Delta_{ijk} = 1$ but different sampling strategy: triplets s.t. $\Psi_{ij} > \theta$ and $\Psi_{ik} < \theta$, which encodes the notion of separating "good" from "bad" pairs.

Optimization. Since it is typically infeasible to enumerate all triplets, this loss function can be optimized through stochastic gradient descent (SGD) [4]. Additionally, since the dimensionality of the features could be large, which would lead to the costly estimation of a large $D \times D$ matrix, we perform a low-rank decomposition $W = U^T U$, where U is a $D \times K$ matrix with $K \ll D$. Note that in this case, the formulation is not convex anymore.

Following straightforward derivations it is possible to show that the learning procedure becomes:

1. Sample a triplet (i, j, k) randomly with $\Psi_{ij} > \Psi_{ik}$.
2. Evaluate its contribution to the loss in Eq.(5): $\mathcal{L}_{ijk}(U) = \max(0, k_U(x_i, x_k) + \Delta_{ijk} - k_U(x_i, x_j))$.
3. If $\mathcal{L}_{ijk}(U) > 0$, perform a gradient step update: $U \leftarrow U + \eta U(x_i \delta^T + \delta x_i^T)$,

where η is a a learning rate, and $\delta = x_j - x_k$.

Note that, after learning, the low-rank decomposition imposes a dimensionality reduction, since $k_U(q, x) = (Uq)^T(Ux)$, which is a dot-product. This leads to a significant reduction of the search cost if the projections are pre-computed for the database (*e.g.* 32K reduced to 8K dimensions in Sec. 6.1).

W is initialized as the identity matrix (or U as a matrix of random numbers drawn from a normal distribution with $\mu = 0$ and $\sigma = 1$, in which case $U^T U \approx I$), and the number of iterations acts as an implicit regularizer (that keeps the solution close to the initial dot product similarity).

5. Task-aware representation

The second contribution is an improved representation built using supervised information of the detection task. More precisely, we propose to build a "probability map" indicating probabilities that a certain pixel contains the prominent object. This is based on a patch-level classifier that has been pre-trained to distinguish between patches from prominent objects and from the rest of the image.

We highlight that probability maps constitute responses of *local* patch classifiers, which are noisy and smooth, and that the response map of explicit object detectors (*e.g.* object banks [19]) would be sparser and more accurate. However, object banks suffer from the same limitations as sliding-window based approaches, while the probability map is fast to compute (just one extra dot product on top of the patch encoding).

Probability maps have been used as an input for object segmentation, to classify super-pixels [5], as the unary potential of a random field [17], or with auto-context algorithms [32]. In our case, we use probability maps directly as an image representation within data-driven detection.

Patch extraction. Patches are extracted densely and at multiple scales within images of the training set, and are associated to a binary label depending on their degree of overlap with the annotated object region. A descriptor is computed for each patch, here a Fisher Vector per patch (as in [5]).

Patch classifier training. Patch-level descriptors and their labels are used to train a linear SVM classifier which assigns each patch to the "object" or "background" class. This classifier introduces the supervised information and makes the representation task-aware.

Probability map computation. An image is represented as follows. First, patches are extracted and described in the exact same way as for training. The patch classifier assigns a score for each patch. These scores are transformed into probabilities at the pixel level [5], yielding probability maps (see Fig. 2, brighter areas correspond to locations more likely containing the object).

Final image representation. To make them comparable, the probability maps are resized to a small fixed-size image (50x50 pixels), ℓ_2 normalized, and the values are stacked into a feature vector. This is our new, task-aware image rep-

resentation, which can also be matched with a dot product.

This representation has several advantages over a task-independent one. First, by nature, this representation captures information about the end task, so images having similar representations are more likely to have similar detection annotations. Second, probability maps are more compact than FV. This means that, despite the extra cost at training time to learn an object classifier, and the small constant cost at test time to compute the map, the smaller dimension of this representation makes retrieval and consequently detection much faster. Also, with its dimensionality (2.5K), if we combine these representations with a task-aware metric, working with the full-rank W is still feasible, which leads to a convex problem. Finally, one can combine several low-level cues (for instance SIFT and color) without increasing the final dimension of the representation by averaging maps computed using different channels [5].

Since probability maps are a strong cue for object location, one may wonder why not using directly a sliding window over the probability map. This is one of the baselines in our experiments. Intuitively, patch level classifiers are far from perfect, but we expect classifier errors to be consistent between similar training and query images. Therefore, even for inaccurate probability maps, the closest maps in the database can help transferring the object location reliably.

6. Experimental validation

We study our approach on three datasets with very different objects to understand its advantages and its limitations: prominent person detection on the Extended Leeds Sports Pose (LSP) dataset, bird detection on the Caltech-UCSD birds dataset, and dog detection on the ImageNet dataset.

Experimental setup. For all the experiments, the task-independent feature chosen is the Fisher Vector (referred to as `FV-SIFT`). Local patches are extracted densely at 5 scales, represented by SIFT (128 dim), and compressed using PCA (64 dim, 32 for LSP). Projected descriptors are used to build a visual codebook of 64 Gaussians. Using it, each image is transformed into a global signature (FV). We only use derivatives w.r.t. the mean. Coarse geometry is introduced by spatial pooling: the image is divided into a regular grid of 8×8 cells (4×4 for metric learning), each bin is described cell a FV, and cell FVs are concatenated [18]. As suggested in [25], we square-root and ℓ_2-normalize FV image signatures. In all experiments, the number of neighbors L are determined from the validation set. From the 3 loss functions described in Sec. 4, option (c) (biased sampling) yields the best results (with θ=0.5, $\eta = 10^{-2}$).

For our task-aware features, the probability maps are built from FVs computed at patch level as in [5]. This essentially follows the same process as explained above but computing one FV per patch instead of aggregating the patch contributions. Probability maps are resized to 50×50 images. The probability maps using FVs computed from low-level SIFT descriptors are denoted `PM-SIFT`. We also consider color statistics [25] as low-level descriptors (`PM-COL`) and the average of both maps (`PM-SIFT+COL`).

	Repr	ML	precision
DDD baseline	`FV-SIFT`	no	34.1
Task-aware DDD	`FV-SIFT`	yes	**43.1**
	`PM-SIFT`	no	39.2
	`PM-SIFT`	yes	**43.0**
Other baselines	`Random DDD`		19.5
	`DPM`		40.3
	`PM-SIFT + SW`		38.6

Table 1. Results of prominent subject detection, in the LSP dataset, that compares task-aware measures (metric learning or ML) and task-aware representations (`PM`) to different baselines.

6.1. Extended Leeds Sport Pose dataset

Dataset and evaluation. We first evaluate our contributions on the *Extended* LSP dataset [14]. It consists of images of persons in unconventional poses. While the database annotations are at the level of body joint locations (knees, elbows, neck, etc), it has been designed so that there is one prominent person (*i.e.* a single annotated subject) per image, which fits well our scenario. We would like to evaluate the prominent subject detection task on these challenging images. To this end, we transformed the annotation, and obtained ground-truth rectangles by taking the bounding rectangle of the body joint annotations. We used the training, validation and test sets as defined in [14]. The quality of the detection is evaluated using the overlap score of Eq. (2), and a detection is considered as correct if the score is above 50% (PASCAL criterion [9]).

Results. First, we study the influence of the task-aware metric, by comparing the two first lines of Table 1, and we see that metric learning brings a significant (+9%) improvement when considering a standard representation. Using the same low level descriptors (*i.e.* SIFT), the task-aware representation `PM-SIFT` still compares favorably to the baseline (+5.1%). Using metric learning on top of `PM-SIFT` still improves (+3.8%), and becomes comparable to the FV with metric learning. Retrieval takes no more than 200ms/image.

We consider 3 additional baselines. First, the state-of-the-art Deformable Part Model (`DPM`) [11, 10], trained with 3 components (+left-right orientations). `DPM` outperforms the task-independent baseline, and is on-par with our task-aware representation. Yet, it is outperformed by the metric-learning approaches. We think that this detector is not well suited to objects presenting such flexible configurations.

We also combine probability maps to a sliding window (SW) process. The rectangle with the best density score (density inside the rectangle minus density outside) is kept. This is referred to as `PM+SW` in Table 1. This baseline

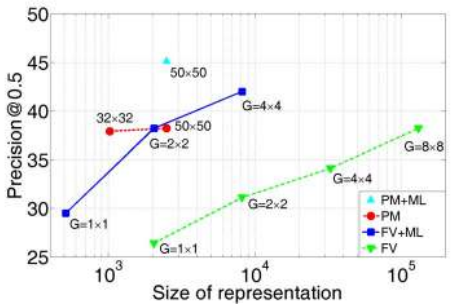

Figure 3. Comparison of DDD for FV (and different spatial pooling G=G_WxG_H) and PM (for different resized map sizes) with and without metric learning.

achieves competitive results (confirming that probability maps are powerful representations), but is outperformed by our best DDD strategies.

Finally, we measure the accuracy of selecting neighbors randomly (`Random DDD`, that validates the number of neighbors). This is to verify that there are no significant annotation bias (an effect sometimes found with large objects). It obtains 19.5%, which is still considerably smaller than any of the proposed methods and baselines.

Dimensionality. For the task aware metric, the low-rank projection reduces the dimensionality from 32K to 8K. The probability map is of size 50×50, thus the dimension is 2500. Other explored parameter combinations (in validation set), and their accuracy vs. dimension trade-offs, are shown in Fig. 3. On top of improved accuracy, we can clearly see the compression offered by the two contributions as a strong advantage in our DDD framework.

Results with color. The previous results use SIFT as low-level descriptor in the probability maps to be comparable to the DPM and DDD baselines that are based on gradients and do not use color. However, as discussed in section 5, additional information about color can be introduced without increasing the size of the representation. When combining low-level SIFT and color descriptors in the probability maps (`PM-SIFT+COL`) and using metric learning, we obtain the best results with 44.5% (some results in Fig. 4).

6.2. ImageNet Dog dataset

Dataset and evaluation. ImageNet dogs is a dataset of dog images used for fine-grained classification purposes [1]. Here, the dog is the prominent object, and we measure the dog detection task, and the effect of our task-aware method on the final classification accuracy. The dataset is composed of 120 different breeds of dogs, making the detection challenging (see Fig. 1). As proposed for the challenge we use 20,580 images for training. As the test annotations are not available, we have split the validation set into 1,000 images that we use as an actual validation set to find the best parameters, and the remaining 5,000 images are used as our test set, for evaluation. We used a generic dog classifier in the task-aware representation, trained using all 120 types of annotations. For detection evaluation, we follow the ImageNet challenge protocol that considers a detection as correct if the predicted bounding box overlaps enough with the ground truth bounding box, or in the case of multiple instances of the same class (which occurs seldomly), with any of the ground truth bounding boxes (*i.e.* at least one object has to be found). We observed that the average overlap between two randomly chosen annotations is already close to 50%. We then decided to report results for a threshold of 70% which is more informative (thus we use $\theta = 0.7$).

	Repr	ML	precision
DDD baseline	`FV-SIFT`	no	35.0
Task-aware DDD	`PM-SIFT`	no	50.2
	`PM-SIFT+COL`	no	51.4
Other baselines	`Random DDD`		14.9
	`Centered R`		31.1
	`DPM`		40.4
	`PM-SIFT + SW`		34.9

Table 2. Results of dog detection, in the ImageNet dataset

Detection results. Results are reported in Table 2. The best results for the `FV-SIFT` baseline are 35.0% and were obtained with large vectors (of 262K dimensions). Our explanation is that with bigger bounding boxes there is less space for diversity in the annotations, which requires more distinctive features. Such a dimensionality leads to prohibitively big projection matrices for task-aware metrics.

The task-aware representation `PM-SIFT` yields 50.2% precision and improves the task independent baseline by +15.2%. Adding color information to the probability map (`PM-SIFT+COL`) yields a further improvement of 1.2% (reaching 51.4%). No further improvement was observed with metric learning on top of these representations.

We consider the same baselines as in the previous section. The sliding window (`PM-SIFT+SW`) is on par with the `FV-SIFT`. Its poor performance could be explained by the following observation: since dogs occupy a large fraction of the image, and they generally do not have a squared shape, the patch classifier training data is more noisy, yielding to probability maps of poorer quality, to which the sliding window is more sensitive. However, if the errors are consistent, DDD is still able to correctly transfer the rectangles even for noisy maps.

DPM[3] performs better, but is still below the proposed task-aware representations (which are much faster). We found it tends to fail for dogs in "non-canonical poses".

The `Random DDD` baseline achieves 14.9%, thus confirming that all methods perform much better than random. One could also ask whether detection is trivial just because dogs seem centered. The `Centered D` baseline predicts

[3]trained with 3 components and 2 orientations, as before

	mAP
No detection (full image)	26.6
DDD (PM-SIFT+COL)	31.2
Ground-truth detection	35.5

Table 3. Fine-grained classification on the Dogs dataset, using detection. Mean average precision is reported.

an object in the center, that occupies 90% of the image (value optimized on validation), and obtains 31.4%. This high number suggests there exists indeed a bias, but the proposed method is still +15.3% above.

Fine-grained classification results. We propose to use localization as an aid for fine-grained classification, by cropping the images using the output of DDD in order to remove noise introduced by the background. We assume that the object location is available at train time (as for DDD), and we train classifiers over the 120 breeds of dogs using the cropped images. At test time, we use the bounding boxes predicted by our system to crop images, and classify the cropped region. For classification, we use FV as a global image representation (using both SIFT and color descriptors with PCA of 64 dim, 256 Gaussians and no spatial pyramid) and linear one-vs-all SVM classifiers. For detection, we use the best DDD method. All parameters are chosen using the validation set of 1,000 images, and classification results are reported for the remaining 5,000 images. As suggested by the challenge organizers, we compute average precision (AP) on individual categories and report the mean average precision (mAP) across all categories.

Classification results are reported in Table 3, and compared to (i) the classification of full images, and (ii) cropping using the ground-truth detections (to measure how far we are from the classification figure of a perfect detection). We confirm that the DDD improves fine-grained categorization and is a few percent points below the perfect detection.

6.3. Caltech-UCSD Birds 200 2011

Dataset and evaluation. Caltech-UCSB 2011 [33] is another fine-grained dataset composed of 200 bird species. We follow the training and test split of [33] (5994 training and 5794 testing images), and use a subset of the training set as validation set (1994 images). We show detection results for generic bird detectors (the bird being the prominent object of each image). The detectors are trained using the 200 types of annotations. The overlap threshold to compute precision is set 70% for the same reason as in the dog set.

Detection results. Results are reported in Table 4 and qualitative results in Fig. 4. For the same reason as in the dogs dataset, we concentrate on the task-aware representations.

In this set, probability maps based on color (PM-COL) yield better results than those based on SIFT (PM-SIFT), probably due to the colorful nature of birds. While PM-COL is on par with the task-independent DDD baseline (FV-SIFT), the combination (PM-SIFT+COL) improves 2.7%.

	Repr	ML	precision
DDD baseline	FV-SIFT	no	24.4
Task-aware DDD	PM-SIFT	no	21.0
	PM-COL	no	23.1
	PM-SIFT+COL	no	27.1
Other baselines	Random DDD		9.02
	Centered D		14.9
	DPM		47.9
	PM-SIFT + SW		12.0

Table 4. Results of bird detection, in the Birds dataset

	Top-1 accuracy
No detection	28.2
Centered D	31.0
DDD (PM-SIFT+COL)	41.9
DPM	42.2
Ground-truth detection	46.3

Table 5. Fine-grain classification on the Birds dataset, using detection. Top-1 accuracy is reported for a global image descriptor.

In this set the difference between DPM and DDD is very large (although DDD is much faster). Our explanation is that birds tend to appear in a few rigid configurations (wings open or closed). Still, the task-aware representations perform largely over the trivial baselines Centered R and Random DDD. The sliding window (PM-SIFT+SW) yields the same behavior as in the dogs set.

Fine-grained classification results. We also conduct a fine-grained classification experiment on this set with the same protocol as before, and measure the impact of detection on the final classification accuracy. For detection, we use the best DDD method (PM-SIFT+COL) but also compare to DPM, as it yielded much better results for detection, and to the (Centered D) baseline, to evaluate against a content-independent cropping strategy. We report the average of the top-1 accuracy across all the categories as this is standard for this dataset. Results are shown in Table 5.

As expected, detection has an impact on fine-grained classification (increase of $>$13%). However, the significant result is that the impact of the DPM and the DDD methods on classification is the same. This looks initially surprising as the difference in detection accuracy is very large. However, the following observation could explain these results. Detection accuracy is evaluated at 70% overlap, thus counting as correct detections only those which are really accurate (note that the PASCAL criterion is 50%). But if we inspect the detection precision at 20% overlap (which corresponds to estimating the object location roughly), the DPM is at 95.0% vs. 99.6% for the DDD. This means there is about 4.6% of the image for which the DPM completely misses the object location. In contrast, the DDD always finds the object location roughly, but not precisely enough for 70% overlap. While a rough detection result can still

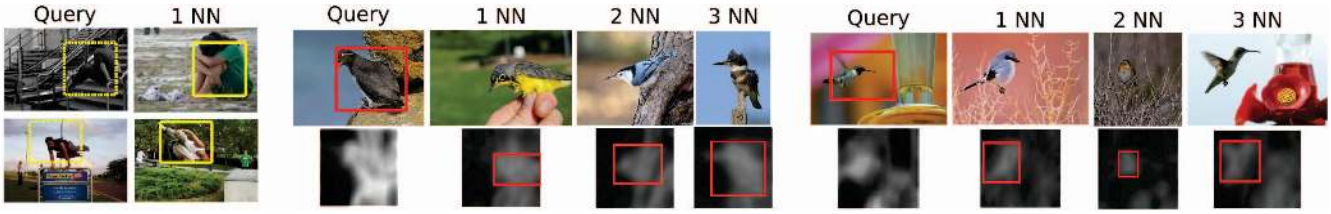

Figure 4. PM-SIFT+COL results. Left: Two examples of the LSP set. Column 1: query image with predicted detection, Column 2: closest neighbor and its ground truth. Right: Two examples of the Birds dataset. The query image and the predicted location. The three closest neighbors with ground truth. Probability maps are also displayed. Predictions might use more neighbors than the ones shown here.

capture information of the class, it is reasonable to expect that most of this 4.6% missed entirely by DPM might translate directly to classification errors. Note the difference in classification accuracy between perfect detection and DPM detection is 4.1%, which approaches the previous number.

7. Conclusion

This paper demonstrates the feasibility of data-driven detection over diverse datasets, and the competitive results it can obtain when enhanced with our two proposed contributions: a task-aware similarity using learning, and a task-aware representation, computed from patch-level object classifiers. Since the proposed method still reduces to finding nearest neighbors at test time using a single feature vector per image, and the two contributions significantly reduce the dimension of the representation, we avoid sliding window search, and the retrieval process is fast (about 200ms per image). DDD compares favorably to a state-of-the art sliding window approach in presence of non-rigid objects. It compares less favorably for rigid objects (as birds) but appears to be good enough as pre-cropping method for fine-grained classification results.

Acknowledgement This work was partially supported by the French ANR project FIRE-ID.

References

[1] http://www.image-net.org/challenges/LSVRC/2012/. 6

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 2

[3] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger. Supervised semantic indexing. In *CIKM*, 2009. 3

[4] L. Bottou. Stochastic learning. In *Advanced Lectures on Machine Learning*, 2003. 4

[5] G. Csurka and F. Perronnin. An efficient approach to semantic segmentation. *IJCV*, 2011. 4, 5

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 2

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007. 3

[8] K. Duan, D. Parikh, D. J. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012. 1

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL voc Challenge 2012 Results. 1, 2, 3, 5

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 1, 2, 5

[11] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/. 5

[12] A. Y. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 2009. 2

[13] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *CVPR*, 2008. 2

[14] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, pages 1465–1472, 2011. 5

[15] D. Kuettel and V. Ferrari. Figure-ground segmentation by transferring window masks. In *CVPR*, June 2012. 2, 3

[16] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*, 2012. 1

[17] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. In *ICCV*, pages 739–746, 2009. 4

[18] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 5

[19] E. P. X. Li-Jia Li, Hao Su and L. Fei-Fei. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010. 4

[20] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE PAMI*, 2011. 2, 3

[21] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*, 2008. 2

[22] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 2

[23] L. Marchesotti, C. Cifarelli, and G. Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *ICCV*, 2009. 1, 2

[24] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, 2010. 3

[25] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010. 3, 5

[26] B. Russell, A. A. Efros, J. Sivic, B. Freeman, and A. Zisserman. Segmenting scenes by matching image composites. In *NIPS09*, 2009. 2, 3

[27] B. Russell, A. Torralba, C. Liu, R. Fergus, and W. Freeman. Object recognition by scene alignment. In *NIPS*. 2008. 2, 3

[28] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Mobile product image search by automatic query object extraction. In *ECCV*, 2012. 1

[29] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little, B. Schiele, and D. Koller. Fine-grained categorization for 3d scene understanding. In *BMVC*, 2012. 1

[30] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010. 2, 3

[31] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE PAMI*, 2008. 2

[32] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *PAMI*, 32(10):1744–1757, 2010. 4

[33] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 7

[34] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 2009. 3

[35] B. Yao, G. Bradski, and L. Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *CVPR*, 2012. 1

[36] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011. 1