Data-Driven Detection of Prominent Objects

Jose A. Rodriguez-Serrano, Diane Larlus, and Zhenwen Dai

Abstract—This article deals with the detection of prominent objects in images. As opposed to the standard approaches based on sliding windows, we study a fundamentally different solution by formulating the supervised prediction of a bounding box as an image retrieval task. Indeed, 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). Our key novelty is to design or learn image similarities that explicitly optimize some aspect of the transfer unlike previous work which uses generic representations and unsupervised similarities. In a first variant, we *explicitly learn to transfer*, by adapting a metric learning approach to work with image and bounding box pairs. Second, we use a representation of images as object probability maps computed from low-level patch classifiers. Experiments show that these two contributions yield in some cases comparable or better results than standard sliding window detectors – despite its conceptual simplicity and run-time efficiency. Our third contribution is an application of prominent object detection, where we improve fine-grained categorization by pre-cropping images with the proposed approach. Finally, we also extend the proposed approach to detect multiple parts of rigid objects.

Index Terms—Object recognition, object detection, metric learning, fine-grained visual recognition, object part localization

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. Note that this is a special case of the object detection problem where we consider a single semantic category of interest, and where we assume that at least one instance of the category of interest is present. This might sound as a strong assumption at first, but prominent object detection is a computer vision task that is required as pre-processing for a large variety of computer vision applications.

Among the many applications of prominent object detection, image thumbnailing [51], [64] or auto-cropping [74] require to locate the foreground object (or the most salient object) to adjust the thumbnail boundaries. In the context of mobile phone applications, such as product search [62], users take pictures of an object of interest which can contain background and possibly other less prominent objects. Localizing the object is often necessary for the next step of the application. Another scenario where this problem is found is fine-grained categorization [17], [42], [63], [72], [73]. 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 we show experimentally). Other examples are commercial scenarios where each object triggers an image capture (e.g., products, or vehicles passing tolls [58]). Nevertheless, our formulation and contributions could be extended to a broader range of localization scenarios and we will discuss possible extensions, such as the localization of object parts.

As all these examples show, the definition of *prominent* or *relevant* object is *application-dependent*, and we assume the object category of interest is defined by a training set with annotated bounding boxes. This departs from objectness methods whose goal is to extract the location of all objects, independently from their semantic categories.

Research in object detection has converged to combining a template descriptor, such as HOG [14] or its extension to deformable parts, DPM [20]), with sliding windows (SW). While these methods have obtained impressive results in several detection scenarios (such as multi-object multi-class ones, see the several PASCAL VOC challenges [19]), one drawback is that they require to classify millions of windows per image. This seems like an overkill in the case of prominent object detection that requires the prediction of a single window. Although methods for accelerating sliding window search have been proposed (such as Efficient Sub-window Search [44], Objectness [3], Selective Search [68], Bing [11] or Edge Boxes [77]), a large number of window descriptors still have to be extracted and classified independently.

In this paper, for prominent object detection, we take a radically different approach and aim at an efficient solution that *extracts a single global feature for the input image and estimates the prominent object location directly from this feature vector*, avoiding sliding windows or extracting candidate windows. Our approach leverages the fact that if descriptors contain spatially-variant information (which can be achieved by spatial pooling), the similarity between those global descriptors provides a strong cue for object location. In other words, *neighbors tend to coincide not only in appearance but also in location* (see Fig. 1). [66, Section V.b] made a similar observation using the raw pixels as features.

- *J.A. Rodriguez-Serrano and D. Larlus are with the Xerox Research Center Europe, France.*
  E-mail: {jose-antonio.rodriguez, diane.larlus}@xrce.xerox.com.
- *Z. Dai is with the University of Sheffield, United Kingdom.*
  E-mail: z.dai@sheffield.ac.uk.

Manuscript received 3 Oct. 2014; revised 4 Nov. 2015; accepted 22 Nov. 2015. Date of publication 16 Dec. 2015; date of current version 12 Sept. 2016.
Recommended for acceptance by C.H. Lampert.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2509988

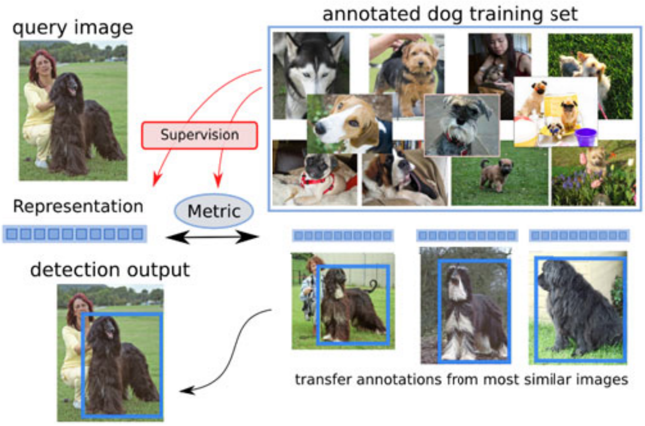
query image
annotated dog training set
Supervision
Representation
Metric
detection output
transfer annotations from most similar images

Fig. 1. Data-Driven Detection: the prominent object of the query is detected using a global representation that is compared to image representations from an annotated set. The nearest neighbors transfer their annotations. We use supervision to learn an image representation and a metric geared toward detection.

This suggests a simple retrieval-based method for prominent object localization: given an input image, find the most similar images from an annotated database (using the global descriptor), and transfer the bounding box of the nearest neighbor. We call this detection-by-retrieval approach *data-driven detection* (DDD). The term *data-driven* often refers to approaches which successfully reduce complex regression problems (e.g. , image annotation, geolocalization) to nearest-neighbor transfer in huge datasets [32], [49], [66], exploiting the phenomenon of the *unreasonable effectiveness of data* [30].

This *data-driven detection* approach exhibits several 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, fixed aspect-ratio, and rigid object assumption of the sliding-window approaches. Finally, as detections are obtained using a global image descriptor, this approach intrinsically leverages context for detection.

However, the data-driven detection idea does not work if applied in the most simplistic setting. First, having images of the object at all the possible locations in the training set is unrealistic for most practical applications, 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 [48], [51], [60], [61], [65] have exploited this “global image transfer”, but they all needed to combine the data-driven idea with more complex and sometimes costly refinements, suggesting that the retrieval step is not sufficient on its own. We believe that this is because these methods use generic (*task-independent*) image representations and similarities that are disconnected from the end-task of detection. Consequently, our two first contributions are improvements to data-driven-detection.

First, we apply metric learning on top of generic image representations to enforce image pairs with similar object layouts to be similar in the feature space, and obtain a **task-aware similarity function**. Second, we propose to build a compact **task-aware representation** that encodes information on the object location to replace the generic image representation.

Both contributions leverage the annotated training set to improve the data-driven detection process, by improving either the global image representation or the similarity measures, connecting them to the detection task. Thus, instead of relying on the assumption that similar images have similar layouts, we actually formulate an optimization problem that *enforces* images with similar layouts to be similar in the feature space. In other words, we *learn* what makes two layouts similar, and use this information for detection.

Experiments indicate that these two contributions significantly improve DDD, and lead to a faster retrieval at test time, as more compact representations are obtained. Despite its conceptual simplicity, data-driven detection leads to an accuracy that on some datasets is on par with the DPM [20].

In a third contribution, we demonstrate the applicability of DDD as a **fast region selector for fine-grained categorization**, reporting improved classification accuracy by applying the classifier on the region predicted by DDD instead of the whole image.

Finally, the fourth contribution is to demonstrate that the method extends beyond rectangular annotations. We discuss how to extend the method to more complex output spaces, and we show results **for efficient detection of rigid object parts**, i.e. localizing several parts of objects simultaneously in an image, still with a single global feature descriptor and a retrieval step. Experiments indicate that it is possible to use the DDD framework for this task by changing the definition of rectangle overlap to a more generic definition of similarity in the output space.

To summarize, our contributions are fourfold: we improve the DDD baseline in two ways i) a task-aware similarity and ii) a task-aware representation, iii) we demonstrate the benefits of DDD in a fine-grained classification scenario, and iv) we extend it to perform efficient detection of object parts.

This paper is an extension of [56]. It is organized as follows. Section 2 reviews previous work. The principle of data-driven detection, and the different contributions we propose are detailed in Section 3. Section 4 presents all our experiments, on detection, fine-grained classification and part detection. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Detection

Standard detection methods combine a window-level representation and a sliding window approach. Among them, HOG descriptors [14] have shown big success for fully rigid objects in multi-class and multi-object settings. DPM [20] 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 [19]. Improvements of the DPM have been proposed, mostly to reduce its heavy computational cost [18], [53], but still rely on sliding windows. Additional representations have been considered for detection, such as Fisher-Vectors [12], or convolutional neural networks [23]. All these methods cast detection as the binary classification of a large number of sub-windows. Sliding windows can be avoided through the use of techniques inspired by Hough voting [7], [46], but they work by aggregating local descriptors. In contrast, in this paper we cast detection as an instance-level *retrieval* problem of global image descriptors.

2.2 Object Proposals

Class-generic methods based on objectness measures [3], [11], super-pixels [68] or edges [77] have been used to generate object proposals. They are used by state-of-the-art detectors to reduce the set of locations that need to be classified. As it significantly speeds up detection, this allows applying sliding windows even to expensive classifiers. Yet, all these methods aim to generalize across categories and fundamentally differ from the task-dependent nature of DDD: they do not incorporate supervised information about the object of interest, and they are not meant to be used as detectors directly, but to output a set of candidate boxes (low precision at high recall).

2.3 Data-Driven Approaches

Data-driven approaches to computer vision [4], [28], [32], [49], [66] exploit the "unreasonable effectiveness" of data [30]: with large enough datasets, some complex prediction problems can be formulated as retrieval problems (or, in other words, k-NN classification ones), which are conceptually simpler. Specifically, given an input image, a straightforward way to predict the output of any computer vision task is to retrieve a near-duplicate from an annotated set, and to transfer its annotation (no matter how complex the annotation space is). This approach had been traditionally avoided in favor of machine learning techniques seeking generalization, as obtaining and scanning a sufficiently large dataset was perceived as impractical. The Tiny Images [66] work is one of the first proofs in vision that tasks such as scene recognition, person localization or colorization can be formulated as a similar image search in a large dataset. Thanks to other large image sets (e.g., ImageNet [16]), and compact state-of-the-art descriptors that enable searching over millions of items in milliseconds [34], data-driven approaches have become feasible and have been applied to image geo-location [32], annotation [49], or scene completion [31].

2.3.1 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 works. Two main strategies exist: transfer at sub-window level and transfer at full-image level.

In the first strategy, approaches still perform sliding window searches and each sub-window is used as a query for which the nearest-neighbor similarity is computed on huge databases of images [66] or sub-images [50]. A similar case is the figure-ground segmentation of [38] 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 (or a large fraction of them) of an image against a database, and pay a complexity price that is at least as big as the one paid by sliding window approaches. The work of [36] proposes a data-driven objectness measure that still requires to extract candidate segments in the image before it scores them.

The second strategy performs transfer at full-image level and exploits the intuition that similar images tend to have similar segmentations [28], [48], [51], [60], [65] or detections [61]. 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. In [51] a saliency classifier is learned on-the-fly for query images using salient and non-salient regions of the nearest neighbors as training data, and the output is refined with a graph-based method. In [48], [59], [60], [65] a Markov random field is instantiated from the transferred segmentations. Similarly, [61] 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 guiding the algorithm, but as the retrieval step is based on *task-independent* features and similarities, this is not sufficient. Methods from this strategy rely on complex models used after the data-driven step. In [38] a global neighbor transfer baseline produces poor results in a multi-object binary segmentation task. Our method belongs to the second strategy, but our novelty is to use supervision in the retrieval step, to build task-aware metrics and representations.

2.4 Metric Learning

Metric learning, distance learning or similarity learning approaches [5], [15], [40], [70] aim at obtaining similarities optimized for tasks such as k-NN classification [15], [70], ranking [5] or regression [71]. In the following, we refer to all these approaches generally as "metric learning" (abusing the language). A significant body of the literature concentrates on finding a linear projection of the data that is optimal with respect to some classification or ranking objective [5], [10], [15], [70]. While in this work we focus on this type of approach, it is worth noting that other alternatives exist, such as boosting [33] or non-linear methods [37].

Specific applications to computer vision include k-NN classification across different domains [41], discriminative retrieval of image annotations [26], face matching [27], person re-identification [76] or image-to-text matching [57]. All these works use categorical labels at the sample level (a sample belongs to a discrete class) or at the level of pairs (a pair of samples is labeled as positive or negative). We are not aware of previous works applying metric learning on object location labels. Here the label space is continuous and multi-dimensional, not categorical.

2.5 Localization for Fine-Grained Classification

Several methods have proposed localization mechanisms to improve fine-grained classification. The methods related to what we propose in Section 4.5 are [21], which uses a shape alignment to localize bird parts, [29] which also uses a nearest-neighbor transfer (but, again, without any task-aware information), and [9] that combines a DPM detector with grab-cut segmentations to localize object parts. These methods then use the extracted regions for pooling better features. The critical difference to our approach is that these papers assume that the location of the object bounding box is known (and used) at test time, while in our case this information is not available, and this is precisely what we try to estimate. It is unclear that [21] and [29] are usable when the location of the object is unknown (which is the more realistic scenario that we consider in our experiments). As [9] could use its DPM component to test this scenario, our experiments also consider DPM for comparison.

Our work also shares similarities with the more recent work of [75]. This work runs a R-CNN-based object detector to find the object location, and uses its nearest neighbors in the appearance space to enforce geometric constraints to the parts. Yet, descriptors from all the proposal regions have to be extracted while our method only extracts global image representations.

3 Data-Driven Detection

3.1 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 [48], [61], [65], data-driven detection can be formalized as follows. We denote by $\{x_i, R_i\}$ a training set of feature-annotation pairs, where $x_i \in \mathbb{R}^D$ is the feature vector of the ith image and $R_i \in \mathbb{R}^4$ the ground-truth rectangle indicating the extent of its object of interest. We encode R_i by the x, y coordinates of its top-left and bottom-right corners, relatively to the image size. We assume a similarity function between features $k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$. 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. A straightforward choice 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 choose $w_i = k_i / \sum_r k_r$. The process is illustrated in Fig. 1.

3.1.1 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 obtained state-of-the-art results in image retrieval [54] and fine-grained classification [25] tasks.

With the use of FVs and Eq. (1) we have a first, simple baseline for DDD. However, as discussed in the introduction, this tends to be insufficient, as both FVs and dot products are task-independent. Next, we present the two task-aware improvements.

3.2 Task-Aware Metric

Our first contribution is a similarity learning algorithm that optimizes a detection criterion.

3.2.1 Definitions

We assume that a similarity function over annotations $\Psi : \mathbb{R}^4 \times \mathbb{R}^4 \rightarrow \mathbb{R}$ 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 intersection-over-union area ratio (e.g., see PASCAL detection challenge [19]):

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

In our case, we resize the images to a standard size before computing this score (i.e. use coordinates relative to the image width and height), so the rectangles are comparable across images of different sizes.

3.2.2 Similarity Function

Our goal is to learn a similarity function $k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ 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:

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

3.2.3 Loss Function

We define a 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 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 dealing 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 *biased sampling*: $\Delta_{ijk} = 1$ and triplets s.t. $\Psi_{ij} > \theta$ and $\Psi_{ik} < \theta$, which encodes the notion of separating "good" from "bad" pairs (with $\theta = 0.5$, $\eta = 10^{-2}$).

3.2.4 Optimization

Since it is typically infeasible to enumerate all triplets, this loss function can be optimized through stochastic gradient

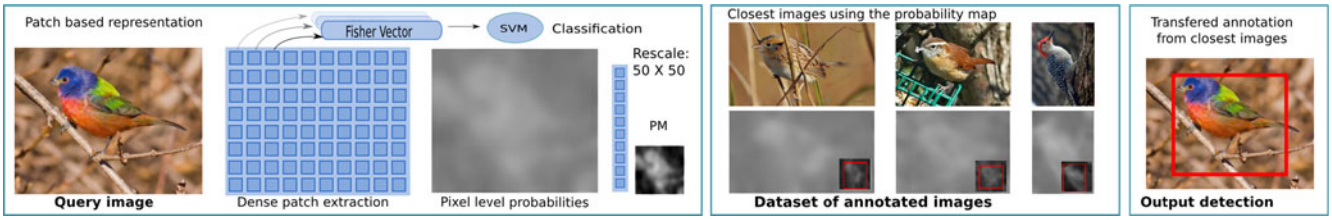
Patch based representation
Fisher Vector
SVM
Classification
Rescale: 50 X 50
PM
Query image
Dense patch extraction
Pixel level probabilities
Closest images using the probability map
Dataset of annotated images
Transfered annotation from closest images
Output detection

Fig. 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.

descent (SGD) [8]. 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 the latter formulation is not convex anymore.

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

1) Randomly sample (i, j, k) 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 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.

In the case of a full-rank matrix, W is initialized as the identity matrix. In the case of a low-rank matrix $W = U^T U$, U is initialized as a matrix of random numbers drawn from a normal distribution with $\mu = 0$ and $\sigma = 1$, in which case $U^T U \approx I$. The number of iterations acts as an implicit regularizer (that keeps the solution close to the initial dot product similarity) [6].

3.3 Task-Aware Representation

In prominent object detection, we assume that all images contain the relevant object of interest. The second contribution also makes use of supervised information of the detection task, but the goal is to improve the image representation itself (rather than the image similarity) to take into account this assumption.

More precisely, we propose to build a "probability map" indicating the probability of a certain pixel to be part of the prominent object. This is based on a patch-level classifier that has been pre-trained to distinguish between patches from prominent objects and patches 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 [47]) would be sparser and more accurate. However, object banks suffer from the same limitations as sliding-window 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 for object segmentation, to classify super-pixels [13], as the unary potential of a random field [43], or with auto-context algorithms [67]. In our case, we use probability maps directly as an image representation within DDD.

3.3.1 *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 [13]).

3.3.2 *Patch Classifier Training*

Patch-level descriptors from the training images 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.

3.3.3 *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 [13], yielding probability maps. In our case, patches are overlapping and multi-scale, which means that each pixel belongs to many patches. Therefore, to compute the score for each pixel we take the weighted mean of the patch probabilities containing the pixel

$$p_x(c) = \frac{\sum_{i|x \in B_i} p_i(c) w_{x,i}}{\sum_{i|x \in B_i} w_{x,i}}, \tag{6}$$

where the weights $w_{x,i}$ are given by the Gaussian Kernel $\mathcal{N}(x|\mu_i, \Sigma_i)$ with μ_i the geometrical center of the patch B_i and Σ_i an isotropic covariance matrix.

Fig. 2 shows examples of obtained probability maps. The brighter areas of the map correspond to locations more likely containing the object.

3.3.4 *Final Image Representation*

To make them comparable, the probability maps are resized to a small fixed-size image ($50 \times 50 = 2.5$ K pixels), ℓ_2 normalized, and the values are stacked into a feature vector. This is our new, task-aware image representation, which can also be matched with a dot product.

This representation has several advantages over a task-independent one. First, it captures information about the end task: by construction, images with similar ground-truth rectangles tend to have similar probability maps. Second, probability maps are more compact than FVs. 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 (2.5 K), 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 [13].

Since probability maps are a strong cue for object location, one may wonder why a sliding window is not directly applied 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 transfer the object location reliably.

3.4 Beyond Bounding Boxes

Most of the experiments in this paper consider that images are annotated with rectangles, as is usually the case in object detection datasets in computer vision. However, the proposed method is applicable to more complex annotations as long as one can define (i) a valid similarity $\Psi : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ between annotations $\mathcal{A}$, and (ii) an averaging function over annotations (i.e., the equivalent of Eq. (1)), if using more than one neighbor. We now provide some examples of possible annotation types that could fit in our framework. We validate experimentally the first case in Section 4.6.

3.4.1 Sets of Rectangles

In Section 4.6 we report experiments on objects annotated by sets of rectangles corresponding to locations of rigid parts. This is a situation which is more challenging than the standard DDD problem on rectangles formulated so far. We provide here the formalism to extend DDD to this type of annotations. Specifically, each image is annotated with a set of "parts" $\mathcal{P} = \{p_1, \ldots, p_T\}$. Each part is defined as $p_t = (r_t, v_t)$, where $r_t \in \mathbb{R}^4$ denotes the bounding rectangle of part t, and $v_t \in \{0, 1\}$ denotes the visibility status (0=invisible, 1=visible). Visibility is incorporated to take into account that some parts might not be visible in the image, e.g., the left wing of a bird cannot be seen from the right side in general.

We define the similarity between two sets of rectangles $\mathcal{P}$ and $\mathcal{P}'$ as

$$\Psi(\mathcal{P}, \mathcal{P}') = \begin{cases} 0 & \text{if } \exists t : v_t \neq v'_t \\ 0 & \text{if } \exists t : v_t = v'_t \text{ and } \Psi(r_t, r'_t) < \theta . \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

In words, we define that two sets of rectangles have similarity one if all the parts have the same visibility, and all visible parts overlap individually by more than a threshold θ. Otherwise the similarity is zero. While other definitions are acceptable, the learning framework described in the last section is directly applicable using the function $\Psi(\mathcal{P}, \mathcal{P}')$ as similarity between annotations.

To deal with multiple neighbors as in Eq. (1), we have to take visibility into account. On top of the non-parametric regression on rectangle coordinates, we also perform the regression on the visibility variable for each part (which amounts to a majority vote).

3.4.2 Pixel Masks

Our approach also extends naturally to pixel masks (a foreground versus background labeling of every pixel), since the similarity between two masks $\mathbf{M}_i$ and $\mathbf{M}_j$ can still be computed as the area overlap ratio of Eq. (2), and the function that predicts the mask $\mathbf{M}$, using the L top-ranked neighbors is straightforward: $\mathbf{M} = \sum_{l=1}^{L} w_{\pi(l)} \mathbf{M}_{\pi(l)}$. The idea of transferring pixel masks from neighbors already appears in [21], [38] and [39], though none of these works attempt to learn a metric between neighbors.

4 Experiments

4.1 Datasets and Evaluation Measures

For the evaluation of the proposed data-driven detector, we consider three datasets with very different objects and properties: the extended Leeds Sport Pose dataset, the ImageNet ILSVRC2012 Dog dataset, and the Caltech-UCSD 200 2011 dataset.

Extended Leeds Sport Pose dataset [35] (ELSP in short) consists of images of persons in unconventional poses. While the database annotations are at the level of body joint locations (knees, elbows, neck, etc.), each image contains one prominent person (i.e. a single annotated subject), which fits well our scenario. We would like to evaluate the prominent subject detection task on these challenging images. To this end, we transform the annotation, and obtain ground-truth rectangles by taking the bounding rectangle of the body joint annotations. We use the training, validation and test sets as defined in [35]. This leads to 4,000 training images, 1,000 validation images, and 5,000 test images.

ImageNet ILSVRC2012 Dog dataset is a set of dog images used for fine-grained classification [1]. Here, the dog is the prominent object, and we measure the dog detection task, and the effect of our task-aware detection 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 set parameters, and 5,000 images that we use as our test set, for evaluation. The detectors are trained using the annotations of the 120 dog species. We also use a generic dog classifier at the patch-level in the task-aware representation, trained using all 120 types of annotations.

Caltech-UCSD Birds 200 2011 dataset [69] is another fine-grained dataset composed of 200 bird species. We respect the training and test split of [69] (5,994 training and 5,794 testing images), and use a subset of the training set as validation set (1,994 images). We show detection results for generic bird detectors (the bird being the prominent object) and fine-grained classification results.

Evaluation measures. First, we measure the expected area overlap of a randomly selected ground-truth bounding box from the training set and one from the validation set (see

TABLE 1
For the Three Datasets, the Top Part Indicates Average Overlaps Used to Choose the Overlap Thresholds that Evaluates the Precision (The bottom part shows results of the control baselines)

Method	ELSP	Dogs	Birds
Average overlap	24.0	45.9	40.7
Selected overlap threshold for precision	50.0	70.0	70.0
Naive baseline: Centered R	26.1	31.1	14.9
Naive baseline: Random DDD	19.5	14.9	9.0
DDD-Selective search baseline	34.4	34.8	16.2
DDD-Objectness baseline	32.3	29.9	20.0

Eq. (2)). The results are given in the first row of Table 1. We observe that in the ELSP dataset the overlap is moderate (24.0 percent) while the relatively high overlap in the Dogs (45.9 percent) and Birds (40.7 percent) datasets indicate that the bounding boxes tend to occupy a big fraction of the image. Consequently, we consider a predicted bounding box to be correct if it overlaps with the ground-truth by 70 percent for the Dogs and Birds datasets, instead of the 50 percent PASCAL criterion, as 50 percent is not meaningful given the measured average overlaps. This is discussed more deeply and additional thresholds are considered in Section 4.4.

4.2 Control Baselines

Before reporting detection and fine-grained categorization experiments on the datasets presented above, we first assess the complexity of the datasets by measuring some "control" baselines. Such baselines are designed to measure the precision obtained by methods that just exploit biases (e.g., centered objects, redundancies in object location), or methods that extract object proposals (indicative for instance of simple textures or easy-to-segment objects). They provide insights about the complexity of the datasets and the tasks, and they serve as an additional basic check.

The **Centered R** baseline predicts a rectangle in the center of the image covering α percent, where α is optimized on the validation set. This baseline scores high for datasets where objects mostly appear in the center.

The **Random DDD** baseline applies the DDD framework to randomly chosen neighbors. This means that the final detections are obtained from rectangles that are transferred from a random set of k images, where k is optimized on the validation set. This baseline would score high for datasets that always present the object in the same location (or the same few locations).

We also discuss two baselines constructed from object proposal methods (see [2], [68] discussed in Section 2). We highlight that none of these baselines can be used directly for DDD as the goal of these methods tends to be detection recall (e.g., high recall in the first 2,000 proposals) rather than precision (which is the goal of DDD). A preliminary experiment confirmed that the top-1 choice yields low precision and poor detection results. Yet we would like to quantitatively show that DDD and such methods are of very different nature.

The **DDD-selective search** baseline extracts the K first object proposals and uses them as if they were the K neighbors of DDD. Note this is non-standard and the only hope for such methods to work is the existence of a single prominent object on a simple background. For that reason, this baseline can be interpreted as a proxy of the fraction of easy-to-detect objects.

For the **DDD-objectness** baseline, we use its intermediate saliency map as an alternative representation for DDD. This baseline would score high if objects are flagrantly salient with respect to the background.

Table 1 reports these baselines on the three datasets. A first observation is that, for all three sets, the Centered R and Random DDD baselines obtain non-negligible values. This indicates that, indeed, there is a bias with respect to the object locations in the way the datasets are constructed (which is not surprising in the case of datasets containing prominent objects). According to these figures, the dataset with highest "center bias" is the ImageNet dogs, and the dataset with most redundancies is ELSP. The results for DDD objectness and DDD selective search indicate that there is a non-negligible fraction of salient or easy-to-find objects in the first two sets—less so in the Caltech Birds dataset. Interestingly, this dataset contains a lot of background clutter and highly textured background (such as vegetation) which makes the detection task more difficult.

However, in the next section we will show that the accuracy of our proposed methods are significantly higher than these numbers, which demonstrates that the datasets are biased but still non-trivial.

4.3 Experimental Validation of DDD

We study our two first contributions on the datasets presented in Section 4.1. We provide evaluation, comparison to control baselines and non-trivial baselines, and analysis of properties and failure cases.

4.3.1 *Experimental Setup*

For all the experiments, the task-independent feature is the Fisher Vector. Local patches are extracted densely at five scales, represented by SIFT (128 dim), and compressed using PCA (64 dim, 32 for ELSP). Projected descriptors are used to build a visual codebook of 64 Gaussians. Using it, each image is transformed into a global FV signature. 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 by a cell FV, and cell FVs are concatenated [45]. As suggested in [55], we square-root and ℓ_2-normalize FV signatures.

For our task-aware features, probability maps are built from FVs computed at the patch level [13]. This essentially follows the same process as explained above but computing one FV per patch instead of aggregating the patch contributions. We use a larger codebook (256 Gaussians, 128 for ELSP) but with no spatial pyramid, and derivatives w.r.t. the mean and the variance. Probability maps are resized to 50 × 50. By default, probability maps using FVs are computed from low-level SIFT descriptors to be comparable to the other methods. As mentioned, probability maps can combine several low-level cues without increasing the final dimension of the representation (by averaging maps) and

TABLE 2
Results of the Different DDD Variants Compared to Several Baselines on the Three Datasets

Method	ELSP	Dogs	Birds
DPM Baseline [20]	40.3	40.4	**47.9**
Sliding window baseline	38.6	34.9	12.0
DDD Plain	34.1	35.0	24.4
DDD + Metric	**43.1**	**52.1**	42.3
DDD + PMap	39.2	50.2	21.0

consequently no additional cost at retrieval time. Therefore, in some cases we will consider maps of color statistics [55], and the average of both maps. The number of neighbors L is determined from the validation set.

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 percent for the ELSP dataset, and 70 percent for the ImageNet dogs and Caltech Birds dataset, as justified in Section 4.1. The results of DDD and its varaiants are shown in Table 2.

Results of DDD. In the three last rows of Table 2, we compare the basic DDD (denoted as DDD plain), DDD with task-aware metric (DDD + Metric) and DDD with a task-aware representation (DDD + PMap) for the three datasets. We observe that the DDD plain represents a surprisingly competitive baseline for its simplicity, yielding 38.3, 35.0 and 24.4 percent precision in the ELSP, ImageNet Dogs and Caltech Birds datasets, respectively. In all three datasets, adding metric learning improves accuracy, by a significant +9 percent in ELSP but by an even larger margin of +15.1 and +17.9 percent in ImageNet dogs and Caltech Birds, respectively. It shows that adding “semantics” of the end-task into the similarity function used in the retrieval process is critical. Regarding the (task-aware) probability map representation, using the same low-level descriptors (i.e., SIFT), improvement over the plain DDD is observed in ELSP (+5.1 percent) and ImageNet dogs (+15.2 percent) but not in Caltech Birds (where there is a loss of 3.4 percent).

Regarding computational speed, these results have not been optimized for accuracy, but in no case detection takes more than 200 ms/image on a single CPU (Intel Xeon 2.4 GHz). Some parameters allow trading off speed for accuracy (such as the rank of the projection matrix) when necessary. Our pipeline inherits from all the efficiency properties of the FV for retrieval, which have been studied in e.g., [54].

4.3.2 Comparison to Control Baselines

Four control baselines were introduced in Section 4.2 to measure the precision obtained by exploiting simple criteria (predicting objects in the center, assigning random scores, or exploiting methods which output “object proposals”).

The DDD plain baseline is comparable to the selective search baseline in the ELSP and ImageNet dogs datasets and outperforms all other control baselines. It outperforms all control baselines in the Caltech Birds dataset. The best DDD variant, DDD with metric learning, always outperforms the best control baseline by a large margin (+8.7 percent in ELSP, +17.3 percent in ImageNet dogs, and +22.3 percent in Caltech Birds). This indicates that the proposed method learns useful information for prominent object detection and does not only exploit (possibly unnoticed) simplicities of the datasets.

TABLE 3
For the Three Datasets, Results of the Task-Aware Representation with and without Color Information

Method	ELSP	Dogs	Birds
DDD + PMap (SIFT)	39.2	50.2	21.0
DDD + PMap (SIFT + Color)	40.3	51.4	27.1

Comparison to alternative approaches. Although prominent object detection differs from detection in its traditional definition, we would like to compare to more traditional detection approaches. We consider two baselines. First, Deformable Part Model (DPM) [20], [24], remains a well-studied and competitive sliding window method. It is trained with three components (+left-right orientations). In the ELSP set, DPM obtains 40.3 percent precision. It is just 1.1 percent better than the DDD with task-aware representation, but is significantly worse than our best DDD with metric learning (43.1 percent). In the Dogs dataset, both the task-aware metric and representation outperform DPM by approximately 10 percent. We believe this result is significant, given the conceptual and computational complexity of DPM with respect to DDD—the latter uses no sliding windows and at runtime only relies on similarity computations between global image descriptors.

In the Birds dataset, however, the best DDD approach is still 5.6 percent below DPM. Although DDD has still advantages over DPM, such as much faster detection, we take this negative result as subject of a deeper study in Section 4.4. Also, we show in Section 4.5 that despite this difference in detection accuracy, when it comes to using those detections as pre-processing for a fine-grained classification task, both systems obtain very similar classification accuracy.

As a second baseline, we combine probability maps to a sliding window process. The rectangle with the best density score is kept. This baseline achieves competitive results (see Table 2), confirming that probability maps are powerful representations, but is outperformed by our best DDD strategies.

4.4 Properties and Limitations of DDD

After quantitative evaluation of the DDD variants, we study in more depth some specific properties of the proposed method as well as shed some light on opportunities for improvement and failure cases.

4.4.1 Results with Color

All previous results use SIFT as low-level descriptor in the probability maps in order to be comparable to the DPM and DDD baselines that are based on gradients and do not use color. However, as discussed in Section 3.3, an interesting property of probability maps is that its size does not depend on the size (or number) of low-level cues used. Thus one can add e.g., color features in the patch classifier, which enriches the expressivity of the representation without increasing its size. Table 3 reports results using both SIFT and color statistics as low-level descriptors.

As shown in Table 3, adding color as a low-level cue brings a slight improvement of +1.1 and +1.2 percent in

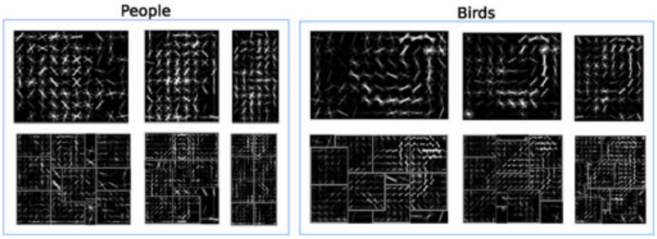

Fig. 3. Root and part filters for each of the three symmetric (six in total) components learnt by DPM for the People and Bird classes, for the ELSP and Caltech-UCSD datasets.

the ELSP and ImageNet dogs dataset, and a significant improvement of +6.1 percent for Caltech Birds. In the latter case, by adding color the task-aware DDD method outperforms the DDD baseline by +2.7 percent (remember with SIFT-only there was a decrease). Fig. 6 shows visual examples of DDD and probability maps (with SIFT + color) on the Birds dataset.

This indicates that the probability map representation with SIFT-only might not be adequate in datasets with very textured objects and background. In such a case a good practice is to add color information—in fact, for the specific case of birds, many species present distinctive color patterns.

The task-aware representation with texture and color can be also combined with metric learning. In the ELPS set, this combination obtains 44.5 percent, which is the best result for the ELSP set achieved in this paper. (for comparison, the sliding window baseline applied on such maps obtains 43.0 percent).

4.4.2 Rigidness of the Dataset

DDD obtains comparable or better accuracy than the DPM detector, which is computationally more expensive, in the ELSP and ImageNet dogs dataset, but DPM is significantly better on the Caltech Birds dataset. One possible explanation is that this detector is very well suited for objects that appear in a small number of canonical rigid poses (the archetypal example is pedestrians), and although it is able to cope with some deformation (as its name highlights) it is challenged when objects are very flexible. In the first two datasets, the prominent subjects tend to be very flexible (especially in ELSP which depitcs humans in challenging sports poses). In contrast, while birds are not fully-rigid objects, they tend to appear in a few set of poses. Our intuition seems to correlate with the visualizations of the learnt DPM models (Fig. 3). In the case of birds, we observe that DPM easily captures three canonical bird poses, explaining the good results. For the person and the dog classes, the models are more fuzzy, the detector has more problems handling these flexible objects.

Fine versus coarse localization. DPM and DDD exhibit very different behaviors when it comes to failure cases. Fig. 4 illustrates this with typical detections. For canonical poses (the one that corresponds well to one of the three (×left-right) models learnt by DPM as illustrated in Fig. 3), the DPM detector produces very well aligned detections, much better than the DDD method (see Fig. 4a). For non-canonical poses, which are the most difficult samples for any detector, DPM outputs little or almost no overlap with the correct bounding boxes, focusing on object parts or being distracted by background clutter (see Figs. 4b, 4c and 4d). In the last example (Fig. 4e), the bird has an unexpected aspect ratio, that confuses DPM. As DDD does not make any assumption on the geometry, its failure cases are more "graceful"—typically, bigger boxes roughly aligned with the object.

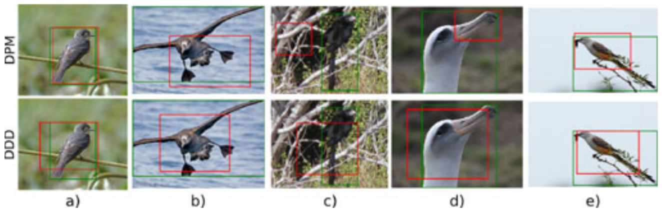

Fig. 4. DPM and DDD exhibit very different behaviors when it comes to failure cases.

This argument is quantitatively supported by the curve on Fig. 5, which illustrates the precision as a function of different overlap thresholds for DPM and DDD (using the best setting of FV and metric learning). We observe that DPM outperforms DDD in general for threshold ≥ 0.5. However, DDD obtains higher precision at lower thresholds. Notably, when the threshold is as low as 0.2, DDD has a precision close to 1, meaning that virtually all failure cases have at least 0.2 overlap with the groundtruth. In contrast the precision of DPM at 0.2 is 95.0 percent, meaning that for 5 percent of the images DPM does not even get close to the groundtruth by 0.2 overlap ratio. In other words, DPM seems to make a 5 percent of severe mistakes while DDD makes an insignificant number of them.

Note that this type of failure is generally preferred, for instance, if the detection output is sent to a subsequent module (e.g., categorization).

4.4.3 Precision as a Function of Object Size

We also study the effect of the object size, for the Birds dataset. Fig. 5 shows the precision of the method as a function of the size of the ground-truth bounding box (expressed as a percent of the image size), and unveils a striking difference between DDD and a detection method such as DPM. While DDD outperforms DPM for moderate and big object sizes (30-90 percent of the image size, almost reaching 80 percent accuracy for images with objects measuring 70-80 percent), it obtains very poor accuracy for small objects. In contrast, DPM has a more balanced precision across all object sizes. This indicates that a limitation of the DDD approach is the ability to deal with very small objects. A similar limitation is observed for the ELSP and ImageNet dogs datasets.

4.5 DDD in Fine-Grained Classification

We propose to use localization as an aid for fine-grained classification, by cropping the images using the output of

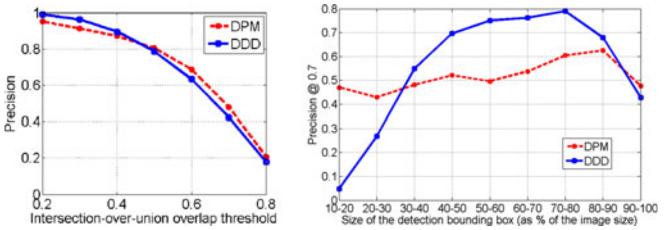

Fig. 5. For UCSD, precision as a function of (left) overlap threshold and (right) object size.

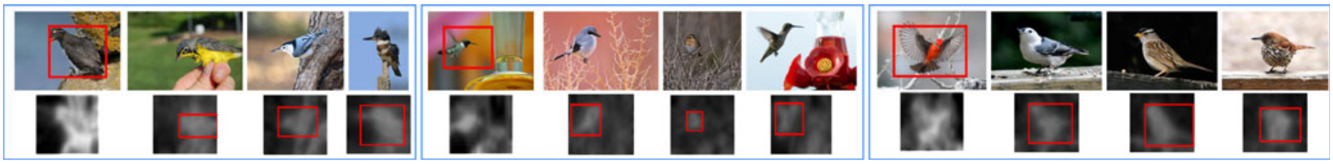

Fig. 6. DDD+PMap (SIFT+Color) results. For each block of four images, the first image is the query, and the three others are the closest three, according to the task aware representation that is shown below. The corresponding object locations are transferred.

DDD in order to remove noise introduced by the background. This can be seen as a way to pool better features from the original images. In the following we show that this leads to improved results on both fine-grained datasets.

Experimental Set-up. For classification, we consider two types of descriptors, both based on FVs. First we use a relatively compact global FV representation with 256 Gaussians and no spatial pyramid (which is what we used in [56]). We also consider a high-dimensional FV representation, with a visual codebook of 1,024 Gaussians and a spatial pyramid that uses three partitions of the image, 1×1, 2×2, and 1×3. FV representations are fed in linear one-vs-all SVM classifiers. In both cases, we use both SIFT and color descriptors, that are combined with late fusion.

For detection, we use the task-aware representation DDD method to select a region of interest in the images, and to crop them accordingly, for all datasets. Note that we preferred those over the task-aware similarity DDD because of their compact nature which is better suited for large scale datasets, and consequently for realistic scenarios.

4.5.1 ImageNet ILSVRC2012 Dog Dataset

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 DDD system to crop images, and classify the cropped regions. 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 [1], we compute average precision (AP) on individual categories and report the mean average precision (mAP) across all categories.

Classification results are reported in Table 4, and compared to (i) the classification of the 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 compare to using the full image, by +4.6 and +4.1 percent respectively for the two descriptors. These results are only a few percent points below the system that uses the perfect detection (i.e., ground truth information).

TABLE 4
Fine-Grained Classification on the Dogs Dataset, Using Detection

Detection Method	Initial res. [56]	New res.
No detection (full image)	26.6	29.8
DDD PMap (SIFT + Color)	31.2	33.9
Ground-truth detection	35.5	37.3

Mean average precision (mAP) is reported.

4.5.2 Caltech-UCSD Birds 200 2011

We experiment on this set with the same protocol as before, and measure the impact of detection on the final classification accuracy. Our system that crops test images with DDD is compared to DPM, as it yielded better results for detection, and to the Centered R baseline (that always selects the center of the image), to evaluate against a content-independent cropping strategy. We report the average of the top-1 accuracy across all 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 percent for both representations). 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, our previous observations from Fig. 5 could explain these results. Detection accuracy is evaluated at 70 percent overlap, thus counting as correct detections only those which are extremely accurate (note that the PASCAL criterion is 50 percent). Detection precision at 20 percent overlap (which corresponds to a rough location estimation) is at 95.0 percent for the DPM versus 99.6 percent for the DDD. This means there is about 5 percent of the images for which the DPM completely misses the object (see the previous discussion related to Fig. 3). In contrast, the DDD always finds the object location roughly, but not precisely enough for 70 percent overlap. While a rough detection can still capture information on the class, we expect that most of this 5 percent missed entirely by DPM might translate directly to classification errors. Note the difference in classification accuracy between the perfect detection (denoted Ground-truth detection) and DDD detection is 4.4 and 6.2 percent for both descriptors respectively. This shows that the proposed classification by detection system goes a nice way towards closing the gap with an oracle using the ground-truth location.

Comparison with State-of-the-art. A comparison with state-of-the-art on the Caltech-UCSD 2011 dataset is difficult, as most fine-grained methods [9], [21], [29] assume that the location of the bird is known, and use the ground-truth bird

TABLE 5
Top-1 Accuracy for Fine-Grained Classification on the Birds Dataset

Detection Method	Initial res. [56]	New res.
No detection (full image)	28.2	32.8
Centered R	31.0	37.1
DDD PMap (SIFT + Color)	41.9	46.7
DPM	42.2	47.6
Ground-truth detection	46.3	52.9

SIFT and color descriptors are used.

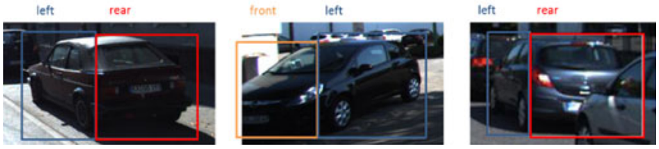

Fig. 7. Examples of vehicle part annotations generated from KITTI. Parts not indicated are invisible ($v_t = 0$).

location at test time as well. To the best of our knowledge, the only works that, at the time of submission, use the same more challenging and more realistic scenario as we do (i.e., the location of the bird is not known at test time) are [52] and [75]. With their proposed Generalized Max Pooling but no spatial pyramid and a smaller vocabulary, [52] reports 33.3 percent top-1 accuracy, which is comparable to our baseline with no location, and is outperformed by the 46.7 percent top-1 accuracy we report when using DDD to predict the object location. Recently, the part-based R-CNN detector of [75] reports 62.75 percent using only a global detector and 73.89 percent with their part-based model. These very good results are obtained extracting CNN features on all the candidate regions of the image.

4.6 Data-Driven Part Transfer

We now consider DDD of multiple object parts simultaneously. Here, instead of a single bounding box, the annotation of an object now contains the labels for individual parts. Each part can be either visible or not, and visible parts are annotated with a bounding box. Therefore, it requires a more sophisticated similarity function as discussed in Section 3.4. In the following, we apply DDD for detection of vehicle parts.

Experimental setup. We use the object detection dataset of KITTI benchmark suite [22]. It contains 3D bounding boxes of vehicles and camera calibration matrices. For the vehicle part detection task, we crop the annotated vehicles around their 2D bounding boxes, and filter out vehicles with too low resolution. In total, 6,647 images are extracted, in which 1,000 images are reserved for test and the rest are used for training. We separate a vehicle into four parts according to its 3D bounding box which are the front, rear, left and right faces (top and bottom faces are discarded because they are rarely visible). The annotation of a vehicle part consists of its visibility and its bounding box. According to the view angle of the camera, the visibility of the parts are estimated, and their 2D bounding boxes are calculated by projecting the corresponding faces of their 3D bounding boxes onto images (see examples in Fig. 7). Therefore, the similarity between the parts of two vehicles can be computed according to Eq. (7), where the overlap score is used for similarity between two visible parts and the threshold θ is 70 percent. In this experiment, FV representations are used for images with a visual codebook of 64 Gaussians and a spatial division of 4×4. A PCA dimension reduction of the FV is applied in this case to accelerate the metric learning, and a full metric W is estimated for reduced image representations. Given a query image, the visibility of vehicle parts and the bounding boxes of visible parts are predicted according to a weighted combination of the L best ranked samples in the training set (see Section 3.4).

TABLE 6
Precision of Vehicle Part Detection, in the KITTI Dataset

Method	Visible parts only				
	av.	right	front	left	rear
Random DDD	20.8	13.5	25.2	24.8	19.7
Image similarity only	78.0	59.1	90.3	95.3	67.4
Individual part learning	**93.6**	**95.6**	**94.9**	**97.3**	**86.5**
the "right face" model	83.3	**95.6**	86.8	81.0	69.5
Proposed	87.7	80.6	91.9	95.8	82.5
Method	Visible and invisible parts				
	av.	right	front	left	rear
Random DDD	-	-	-	-	-
Image similarity only	85.3	88.0	85.9	84.4	83.0
Individual part learning	79.1	84.7	73.7	76.5	81.5
the "right face" model	83.5	84.7	83.8	84.8	80.6
Proposed	**90.9**	**93.2**	**91.6**	**90.1**	**88.5**

Results. DDD algorithms are applied following two scenarios. The first scenario only predicts visible parts, knowing the visibility of the query vehicle, which is an idealistic setting. A prediction is considered correct if the overlap score is above 70 percent. The second scenario predicts both visible and invisible parts, which is more realistic. A prediction is considered correct only if the predicted visibility is correct and the overlap score of visible parts is above 70 percent. The precision of vehicle part detection in both scenarios is reported in Table 6.

To assess the bias of object part locations in the dataset, Random DDD (see Section 4.1) is applied. Its result indicates that there is a bias for object part locations, but its performance is significantly lower than any DDD algorithms. Then, with a task-independent DDD baseline ("image similarity only"), one obtains on average 78 percent for visible parts only and 85.3 percent for both visible and invisible parts. By applying the standard metric learning for individual parts, the precision for visible parts is significantly improved (+15.6 percent on average), while the precision for both visible and invisible parts decreases by -6.2 percent, as the metric learning does not take into account the visibility of parts. Additionally, we check against a baseline that learns a metric in the DDD framework with the annotations of a single part (e.g., right face), and this metric is then used to transfer all the parts at test time. We show results for the right face, which works best. As expected, the precision is lower than using individual metrics (-10.3 percent on average), but is better than task-independent DDD (+5.3 percent on average) for visible parts.

The proposed method (learning a single metric, see Section 3.4) obtains on average 87.7 percent for visible parts only and 90.9 percent for both visible and invisible parts. It is worse than learning individual metrics (-5.9 percent on average) for visible parts only, but is significantly better (+11.8 percent on average) for both visible and invisible parts. The conclusion is interesting: when the visible parts are known at test time, the best results are obtained by learning a standard metric per individual part (as in the rest of the paper). However, this corresponds to an unrealistic setting, and if the parts visibility is unknown a priori (i.e. the realistic case), learning a single metric with a loss function taking into account all the parts jointly and their

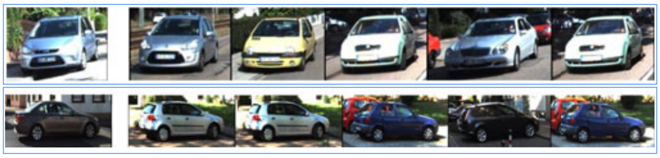

Fig. 8. Vehicle pose retrieval. Left: query image, right: five nearest neighbors using the learned metric.

visibility, as proposed, obtains the best results. The reason is that if some parts could be missing at test time, it is crucial to add this information during learning. This results in an effective way of extending DDD for the joint localization of multiple parts.

An interesting by-product of this experiment is that the learned metric acts as a robust cue for pose retrieval: similar images in the new projected space tend to represent vehicles in the same pose (see Fig. 8). This is not the case of the task-independent metric, which tends to favor visually similar vehicles (not necessarily in the same pose).

5 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 200 ms per image). DDD compares favorably to a standard 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 tasks. Future work will aim at investigating the use of deep learned representations.

Acknowledgments

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

References

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

[2] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 73–80.

[3] B. Alexe, T. Deselares, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[4] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1092–1099.

[5] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger, "Supervised semantic indexing," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 187–196.

[6] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger, "Learning to rank with (a lot of) word features," *Inform. Retrieval*, 2010, pp. 291–314.

[7] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recog.*, vol. 13, no. 2, pp. 111–122, 1981.

[8] L. Bottou, "Stochastic learning," in *Proc. Adv. Lectures Mach. Learn.*, 2003, pp. 146–168.

[9] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 321–328.

[10] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learn. Res.*, vol. 11, pp. 1109–1135, 2010.

[11] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3286–3293.

[12] R. G. Cinbis, J. Verbeek, and C. Schmid, "Segmentation driven object detection with Fisher vectors," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2968–2975.

[13] G. Csurka and F. Perronnin, "An efficient approach to semantic segmentation," *Int. J. Comput. Vis.*, vol. 95, no. 2, pp. 198–212, 2011.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comp. Soc. Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 886–893.

[15] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 209–216.

[16] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[17] K. Duan, D. Parikh, D. J. Crandall, and K. Grauman, "Discovering localized attributes for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 3474–3481.

[18] C. Dubout and F. Fleuret, "Exact acceleration of linear object detectors," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 301–311.

[19] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL VOC Challenge 2012 Results,"Available: http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2012/index.html

[20] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[21] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1713–1720.

[22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 580–587.

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

[25] P.-H. Gosselin, N. Murray, H. Jégou, and F. Perronnin, "Revisiting the Fisher vector for fine-grained classification," *Pattern Recog. Lett.*, vol. 49, no. 1, pp. 92–98, Nov. 2014.

[26] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 309–316.

[27] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? metric learning approaches for face identification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 498–505.

[28] R. Guo and D. Hoiem, "Beyond the line of sight: Labeling the underlying surfaces," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 761–774.

[29] D. Haase, E. Rodner, and J. Denzler, "Instance-weighted transfer learning of active appearance models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1426–1433.

[30] A. Y. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Intell. Syst.*, vol. 24, no. 2, pp. 8–12, 2009.

[31] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[32] J. Hays and A. A. Efros, "IM2GPS: Estimating geographic information from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[33] T. Hertz, A. Bar-Hillel, and D. Weinshall, "Learning distance functions for image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. II-570–II-577.

[34] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[35] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1465–1472.

[36] H. Kang, A. Efros, T. Kanade, and M. Hebert, "Data-driven objectness," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 189–195, Jan. 2015.

[37] D. Kedem, S. Tyree, F. Sha, G. R. Lanckriet, and K. Q. Weinberger, "Non-linear metric learning," in *Proc. Adv. Neural Inform. Process. Syst*, 2012, pp. 2573–2581.

[38] D. Kuettel and V. Ferrari, "Figure-ground segmentation by transferring window masks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 558–565.

[39] D. Kuettel, M. Guillaumin, and V. Ferrari, "Segmentation propagation in ImageNet," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 459–473.

[40] B. Kulis, "Metric learning: A survey," *Found. Trends Mach. Learn.*, vol. 5, no. 4, pp. 287–364, 2013.

[41] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1785–1792.

[42] 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 *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 502–516.

[43] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 739–746.

[44] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2129–2142, Dec. 2009.

[45] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 2169–2178.

[46] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 259–289, 2008.

[47] L.-J. Li, H. Su, E. P. Xing, and L. Fei-Fei, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Proc. Adv. Neural Inform. Process. Syst.*, 2010, pp. 1378–1386.

[48] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, Dec. 2011.

[49] A. Makadia, V. Pavlovic, and S. Kumar, "A new baseline for image annotation," in *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 316–329.

[50] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMS for object detection and beyond," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 89–96.

[51] L. Marchesotti, C. Cifarelli, and G. Csurka, "A framework for visual saliency detection with applications to image thumbnailing," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 2232–2239.

[52] N. Murray and F. Perronnin, "Generalized max pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2473–2480.

[53] M. Pedersoli, A. Vedaldi, and J. Gonzàlez, "A coarse-to-fine approach for fast deformable object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1353–1360.

[54] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3384–3391.

[55] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.

[56] J. A. Rodríguez-Serrano and D. Larlus, "Predicting an object location using a global image representation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1729–1736.

[57] J. A. Rodríguez-Serrano, A. Gordo, and F. Perronnin, "Label embedding: a frugal baseline for text recognition," in *Int. J. Comput. Vis.*, vol. 113, no. 3, pp. 193–207, 2015.

[58] J. A. Rodríguez-Serrano, H. Sandhawalia, R. Bala, F. Perronnin, and C. Saunders, "Data-driven vehicle identification by image matching," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2012, pp. 536–545.

[59] A. Rosenfeld and D. Weinshall, "Extracting foreground masks towards object recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1371–1378.

[60] B. Russell, A. A. Efros, J. Sivic, B. Freeman, and A. Zisserman, "Segmenting scenes by matching image composites," in *Proc. Adv. Neural Inform. Process. Syst.*, 2009, pp. 1580–1588.

[61] B. Russell, A. Torralba, C. Liu, R. Fergus, and W. Freeman, "Object recognition by scene alignment," in *Proc. Adv. Neural Inform. Process. Syst.*, 2008, pp. 1241–1248.

[62] X. Shen, Z. Lin, J. Brandt, and Y. Wu, "Mobile product image search by automatic query object extraction," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 114–127.

[63] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little, B. Schiele, and D. Koller, "Fine-grained categorization for 3D scene understanding," in *Proc. Brit. Mach. Vis. Conf.*, 2012.

[64] J. Sun and H. Ling, "Scale and object aware image thumbnailing," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 135–153, 2013.

[65] J. Tighe and S. Lazebnik, "Superparsing: Scalable nonparametric image parsing with superpixels," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 352–365.

[66] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large dataset for non-parametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.

[67] Z. Tu and X. Bai, "Auto-context and its application to high-level vision tasks and 3D brain image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1744–1757, Oct. 2010.

[68] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, "Segmentation as selective search for object recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1879–1886.

[69] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, Tech. Rep. CNS-TR-2011-001, 2011.

[70] K. Weinberger and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009.

[71] K. Weinberger and G. Tesauro, "Metric learning for kernel regression," in *Proc. 11th Int. Conf. Artif. Intell. Statist.*, 2007, pp. 608–615.

[72] B. Yao, G. Bradski, and L. Fei-Fei, "A codebook-free and annotation-free approach for fine-grained image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 3466–3473.

[73] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1577–1584.

[74] M. Zhang, L. Zhang, Y. Sun, L. Feng, and W. Ma, "Auto cropping for digital photographs," in *Proc. Int. Conf. Multimedia Expo*, 2005, p. 4.

[75] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[76] W.-S. Zheng, S. Gong, and T. Xiang, "Person re-identification by probabilistic relative distance comparison," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 649–656.

[77] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.

Jose A. Rodriguez-Serrano received the degree in Physics in 2003 from the University of Barcelona and the PhD degree in 2009 from the Universitat Autonoma de Barcelona. Since May 2014, he has been an area manager in the Machine Learning for Services group, XRCE, after being a permanent research scientist since 2010 at the Computer Vision Group. Previously, he had been a postdoctoral fellow at the University of Leeds and Loughborough University. His main interests are solving problems of industrial interest through the use of state-of-the-art machine learning and computer vision research, and is fascinated by the challenge of making research outcomes simple to use for non-experts.

Diane Larlus received the MS degree in image, vision, and robotics from the National Polytechnic Institute of Grenoble (INPG), in 2005, and obtained the PhD degree in the LEAR group of the INRIA-Grenoble laboratory, in 2008. She is a senior research scientist at the Xerox Research Center Europe (XRCE), in Grenoble, France. In 2007, she was awarded the JSPS grant to collaborate during the summer with the JRL group, at the CNRS/AIST laboratory of Tsukuba, Japan. From 2008 to 2010, she has been a postdoctoral fellow in the Multimodal Interactive Systems group, TU Darmstadt, Germany. Her research focuses mainly on computer vision and on statistical learning methods applied to scene understanding.

Zhenwen Dai received the BSc degree in computer science from the Zhejiang University, China, in 2007, and the MPhil degree in computer science from The University of Hong Kong, Hong Kong, in 2009, and the doctoral degree in computer science from the Goethe-University Frankfurt, in 2013. He is currently a postdoctoral research associate at the University of Sheffield in the field of Machine Learning and Computer Vision.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.