Nuances in visual recognition
Gavves, E.

Citation for published version (APA):
Gavves, E. (2014). Nuances in visual recognition

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

“It is not the strongest or the most intelligent who will survive but those who can best manage change.”

– Charles Darwin (1809-1882)
Figure 28: Examples of fine-grained sub-classes for the Birds and Dogs datasets. Note the difficulty of recognizing these categories in a finer detail. (a) All four birds belong to different sub-classes, although some of them look very similar. (b) Dogs appear in all kinds of position, poses and scales. Based on example images like these, fine-grained categorization tries to discover which fine-grained species each image belongs to. Rather than directly trying to localize parts (be it distinctive or intrinsic, see text), we propose to first roughly align the objects based on their global shape, ignoring the actual fine-grained category. After aligning the object, we then proceed with consistent partitioning, arriving at successful classification.

In this part of the thesis we study situations, where discovering distinctive visual nuances are not just important, but vital for successful classification and visual understanding. We, therefore, elaborate on the task of fine-grained categorization. Fine-grained categorization deals with object categories that belong to a common super category, for example bird species or dog breeds.

5.1 Introduction

According to cognitive psychology, fine-grained categorization of images, like the ones in Figure 28, relies on identifying small differences in appearance of specific object parts [111]. Humans learn to distinguish different types of birds by addressing the differences in specific details. Recent works in computer vision have verified this mechanism [11, 26, 40, 164, 165]. The same holds for car types [36], aircraft types [84] and dog breeds [64, 80]. Active learning methods have been proposed to extract attributes [38], volumetric models [40] or part models [18]. They require expert-level knowledge at runtime. In contrast, we aim for fine-grained image categorization from training example images, with no interaction other than the fine-grained label.

Various methods learn what details to focus on for fine-grained categorization. While good results have been obtained by relying on high dimensional template matching procedures [158], parts are adopted as the natural template [165]. Yet, it remains unclear to what degree is the ability to accurately localize corresponding locations over object instances important, counterbalanced by the ability of capturing detailed information from raw visual data? While often these go hand in hand, e.g., when using templates, we defend the view that actually it is the latter that matters most. Therefore, we argue that precise localization is not always necessary. Rough alignments suffice, as long as one manages to capture the distinctive details in the appearances.
Localizing consistent locations on instances of certain object categories is strongly related to part learning. Parts are divided into intrinsic parts, i.e., semantic parts that are shared by all (or at least most of) the sub-classes, as in [18, 80], such as the head of a dog or the body of a bird, as opposed to distinctive parts, as in [157, 158] specific to a few sub-classes. The large variability in poses and appearances renders the clean detection of intrinsic parts difficult. In contrast, distinctive parts are destined to be found on few sub-classes only. They are more consistent in appearance, as the distinctive detail is better tailored to be detected on few sub-classes. Still, the number of sub-class specific parts soon becomes huge, each trained on a small number of examples. This limits the robust capturing of all viewpoints, poses and condition changes. Hence, detecting parts, be it intrinsic or distinctive, both have their difficulties in the learning phase.

Rather, we propose to roughly localize distinctive details by first aligning the objects. This alignment is rough and insensitive to most appearance variations. Rough alignment is not sub-class specific, thus the object representation becomes independent of the number of classes or training images [158, 159]. In essence, rough alignment rests on the assumption that the sub-classes share a rough shape.

Our first contribution is based on the observation that all sub-classes belonging to the same super-class share a similar global pose. Within that pose similar visual properties are found on similar locations. Therefore, it is effective to align objects, as we will pursue. In supervised alignment, annotated details are transferred from training images to test images. In unsupervised alignment, we use alignment to delineate corresponding regions on objects, so that they can be used in the differential classification.

Our second contribution is based on the observation that starting from rough alignments small appearance perturbations will become noticeable even between very similar objects, due to common image deformations such as small translations, viewpoint variations and partial occlusions. Descriptors that are precise, yet sensitive to common image transformations, like intensity SIFT [81] or kernel descriptors [14], used in prior works [40, 157, 164] are therefore likely to be a sub-optimal choice for a description of parts. We prefer to use localized, color Fisher vectors that are originally developed for global image classification [63, 104]. As our experiments indicate, they are more suited than spatial/matching based intensity features not only for describing roughly localized information, but even ground truth parts.

As part of our third contribution, we reveal the significance of segmentation in fine-grained categorization. As fine-grained categorization implies that all sub-categories are usually found in common premises, segmentation isolates the pixels relevant to fine-grained details only. Our experiments show that accurate segmentation makes a substantial difference in the recognition of fine-grained sub-categories. What is more, we demonstrate that segmentation based free-form parts, allow for higher precision, as compared to traditional rectangular templates.

Our forth contribution is to present a methodology for performing fine-grained categorization with minimal human interaction. Where often bounding boxes are required at runtime, see [26, 44, 165], we demonstrate how to obtain a similar support region with simple, inexpensive means. The amount of human interaction is thus limited to only providing training images, without the need for bounding boxes.

Last, we include a qualitative analysis that outlines the limitations of visual features. Where visual features extracted from the fine-grained object fail to discern between species, possibly due to almost identical appearance, one could attempt to analyze the environment, as [35] would argue. We conclude with an experiment, where we answer the question of what makes a bobolink a bobolink, finding out that advanced, orderless, features, such as Fisher vectors, operate as a spatial hashing function, that builds correspondences between spatial details and certain feature dimensions.
This chapter is an extension of our previous work [44]. Compared to our earlier version, we present a richer related work section, and we enrich our methodology by i) extending the types of fine partitions and ii) alleviating the bounding box requirement at runtime. Furthermore, we extend the experimental section by including seven more experiments, qualitatively and quantitatively evaluating all extensions on the challenging Birds [148] and Dogs [64] datasets.

We proceed with presenting a list of related works on fine-grained categorization in the Section 5.2. In Section 5.3 we describe the proposed method, including the localization, the extraction and the description of alignments. Experiments are presented in Section 5.4 and we summarize our conclusions in Section 5.5.

## 5.2 Related Work

We organize our discussion on related fine-grained categorization works by the vision tasks involved: localization, partitioning, and description. Within each task we organize the works by the amount of required human intervention.

### 5.2.1 Localization

Many works in fine-grained categorization assume that the (bounding box) location of the object is available, both at training and test phase, see [11, 26, 37, 44, 59, 155, 157–159]. Knowing a priori the location of the fine-grained object allows to focus on the detection and description of the fine-grained details only. Hence, the above works report the highest recognition rates in the literature, although it was shown by [158] that a bad bounding box can be more harmful than having no box at all. In the current work we localize fine-grained objects, without requiring a bounding box.

Others require annotations only during training. Inspired by the poselets of [16], [40] use volumetric primitives, the “birdlets”, parameterized to reflect the 3-D geometry of the body and head of birds, resulting in pose normalized representations. Since birdlets require expensive 3-D ground truth annotations, they are limited to small datasets. Therefore, [164] propose to first employ simpler to detect 2-D poselets, which are then warped in order to arrive at a consistent, pose-normalized representation. Others require only bounding boxes for the location of the fine-grained objects during training. [18, 147] employ deformable part models [41] for detection, showing, however, that user feedback is necessary to improve accuracy. In contrast to the above works, we localize fine-grained objects without requiring anything but the class label for training.

Others working under such conditions proceed with fine-grained categorization, without expecting any information regarding the location of the fine-grained objects, neither during training nor during testing. While [114] focus on image-level descriptions, purposefully ignoring the spatial aspect, the main focus has been to discover the object’s location in an unsupervised manner, usually applying image-level segmentation like in [95], or co-segmentation methods like in [25, 27]. We rely on segmentation as well.

We propose a multi-functional approach that performs accurate fine-grained categorization, when bounding boxes are i) provided during training and testing, ii) only during training, using supervised object detectors like [41] at test time or iii) not provided at all, using unsupervised object proposals like [86, 136] both at training and test time. In the latter case we report competitive recognition rates that often outperform methods requiring bounding boxes. Last, we evaluate the importance of accurate segmentation.
5.2 Related Work

5.2.2 Partitioning

When classifying different bird sub-classes, like telling the Forster’s Tern apart from the Least Tern, see Fig. 28, one probably needs to discover details such as their beak color patterns. Since consistently localizing such details is assumed to be crucial, a large part of the fine-grained literature has put considerable effort in this task, see [11, 26, 40, 80, 155–159, 164, 164].

Some methods focus on an active learning approach for detecting locations. [147] consider user clicks, guiding the machine to pose the most informative question to the user, while [18] propose online supervision to learn better part models. Given ground truth part annotations, part sharing between classes was shown by [80] to result in accurate dog breed recognition. Going one step further, [155] demonstrate excellent results for fine-grained categorization, assuming that ground truth part annotations are available also at runtime. We do not require part annotations at runtime.

The majority of works, however, targets towards automatic partitioning. [159] use randomized trees to mine discriminative features. In [158] the same authors propose to randomly generate thousands of templates, which after being convolved with the unseen images lead to very high-dimensional representations. Extracting unsupervised templates, which take into account part appearance, co-occurrence and diversity, was shown by [157] to deliver excellent results in several datasets. Inspired by the partial object model of [12], [40] and [164] consider the head of a bird as most discriminative, using it to perform recognition. Moreover, [11] showed that ground truth part annotations can be used for designing intricate features specific to certain sub-categories, arriving at excellent recognition rates. And recently, [26] and [165] proposed to employ modified deformable part models [41], to detect consistent fine-grained parts that allow for pose-normalized representations.

Similar to the above works, we detect interesting object locations for discriminating between sub-species. Different from the above works, we do not aim at directly localizing individual parts. Instead, we propose to first align the object as a whole. Based on this alignment, we then derive a small number of partitions. Although our alignments and the subsequent partitionings can benefit from supervision during training, we show that obtaining them in an unsupervised manner is feasible, leading to high recognition in fine-grained categorization that outperforms the state-of-the-art.

5.2.3 Description

For the description of features several possibilities have been explored in the literature, some of them requiring user assistance, while the majority is fully unsupervised.

Methods that propose user-assisted features mainly focus on interpretable attributes. Discovering discriminative, user-accredited attributes, e.g. whether a bird has spots or not, has been repeatedly explored by [19, 99]. In a similar manner, [38] detect mid-level attributes, which are, however, location and not image-level specific. Since attributes need to be interpretable to make sense, human labor and often expert knowledge is required, rendering these approaches useful for small datasets only as in [38]. In our work we do not attempt to represent fine-grained objects in terms of mid-level features or attributes.

Most works in the fine-grained categorization literature do not require human-interpretable features. Raw features, such as intensity SIFT proposed by [81] or kernel based descriptors proposed by [14] have shown to be good choices in describing the distributions of low level appearance details, such as edges or color [40, 157]. However, being sensitive to misalignments renders them less suited for objects that are distorted in the presence of common image deformations. To cope with such misalignments, feature encodings have also been proposed. Locality-constrained linear coding in [159], bag-of-words in [164] and Fisher vectors in [26, 27, 114] were shown to describe
fine-grained categories accurately. For an excellent review on how to adapt Fisher vector for
fine-grained categorization we refer to [49]. Furthermore, [11] showed that supervised features
trained to be discriminative for pairs of classes achieve state-of-the-art results. And [37] showed
that employing a deep learning architecture specialized to fine-grained subcategories arrives at
remarkable recognition rates, at the expense of requiring additional images for feature learning.
Here, we also propose to use unsupervised features, more specifically Fisher vectors [104].
Different from most previous works, we extend Fisher vectors to operate not only as global,
object-level representations, but also to encode the localized appearance of object parts.

Another interesting aspect of the description of object locations is the exploitation of domain
specific, low-level appearance, such as color. Intuitively, in fine-grained sub-categories of the
natural world, such as birds species, see [148] or dogs breeds, see [64], color is bound to have
a great impact in telling sub-categories apart. Surprisingly enough, the recent fine-grained
literature [40, 158] often focuses on more traditional color based descriptors as found in [126]
rather than state-of-the-art solutions, see [63]. We evaluate and highlight the potential of color in
fine-grained categorization, when advanced color descriptors are considered.

5.3 LOCAL ALIGNMENTS AND PARTITIONS

Within a fine-grained categorization setting we assume an image I contains an object belonging
to one of the 1, ..., K sub-categories of interest. Naturally, there might be several other objects
present in the image and not just the fine-grained object. Furthermore, we do not restrict the
location and scale of the fine-grained object. Although in a fine-grained categorization setting
these problems are often evaded by assuming that bounding boxes are provided by humans at
query time like in [11, 26, 44, 158], in real world scenarios it is not always realistic to expect
such user input. Therefore, localization of the object of interest needs to precede any further
fine-grained analysis regarding the specific sub-category that is depicted. For localization, we
propose to use object detection as a soft prior for segmentation, to avoid important details to be
missed.

The localization provides a local frame of reference that serves to identify the spatial properties
of the object. When we identify a local frame of reference in an image, consistent with other
local frames of reference in other images, then we call the image aligned. Consistent means that
corresponding parts are found in corresponding locations, when expressed with respect to their
frame of reference.

By design we opt for finding the parts consistently, at the cost of less precise detections,
accepting the small drift in part appearance that might occur. To avoid being oversensitive to such
drifts, we choose our supervised and unsupervised alignments to be rough but consistent, rather
than precise but unstable. Given the rough nature of our alignments, we show that orderless,
powerful features are the preferred choice.

5.3.1 Localization

Why not an object detector?

In order to discover the spatial support of an object the apparent choice is to employ an object
detection algorithm, see [41, 86, 136, 143]. In that case, we predict the best possible bounding box
that surrounds the object of interest as tightly as possible. A successful detection D is evaluated
with respect to the amount of overlap between the predicted bounding box and the ground truth bounding box $G$

$$\text{overlap} = \frac{D \cap G}{D \cup G}. \quad (5.1)$$

The overlap penalizes both inclusion of extra background and the exclusion of foreground. Since detection is difficult by nature, usually some error margin is allowed. This error margin is expressed as a minimum overlap threshold, above which detection is considered to be correct. State-of-the-art challenges [39] set this threshold to 50%. The design of the overlap measure in eq. (5.1), therefore, suggests that detections should minimize the amount of the background in the detection $D$, even if some foreground is missed.

This setup, however reasonable for object detection, may cause problems to the subsequent segmentation required for fine-grained categorization, see [18,147]. To illustrate with an example, having a box overlapping 50% with the object of interest suffices for an object detector. However, 50% of overlap also implies that a large chunk of the object’s body may be missed, thus potentially losing the crucial details that make the difference between, e.g. the “Magnolia Warbler” and the “Myrtle Warbler”. Furthermore, performing segmentation for all the bounding box candidates returned by state-of-the-art object detectors, like [31,136], would be computationally challenging. To this end we propose to alter the way traditionally object detectors are employed and use them as soft priors for segmentation.

**Objectmaps**

Given a detection algorithm, we expect a sizable number of bounding boxes $\{D_i\}$ that indicate potential existence of the object of interest in the respective image region. While some detectors, e.g. [136], are designed to return several box candidates, others, e.g. [41], are parameterized to return only few. For the latter ones we set their reliability threshold sufficiently low, thus acquiring several promising candidates as well. As explained above, we do not consider these bounding boxes to be accurate enough to be trusted for as is. However, we do consider them accurate enough as soft voters, that collectively return the confidence that the pixel $p$ lies on an object, that is

$$o(p) = \frac{\sum_i D_i(p)}{Z}, \quad (5.2)$$

where $D_i(p) = 1$ when the $i$-th bounding box contains the pixel $p$ and $Z$ is a normalization constant such that $\max o(p) = 1$. We will refer to the spatial prior $o(p)$ as objectmap.

Not all bounding boxes returned by object detectors are relevant. We therefore employ filter functions to prune the ones that are unlikely to cover part of the object. The first filter relates to the size of the bounding boxes. As observed by [23,136], the size of the relevant bounding boxes strongly depends on the specific dataset at hand and a minimum bounding box size is usually enforced. We discard the bounding boxes with unlikely geometries according to the training images, e.g., too extreme width-to-height aspect ratios. Although some boxes will incorrectly be discarded, the rough location estimation depends on the collective power of several bounding boxes. Hence, missing a few relevant ones is not critical, as long as the majority concentrates around the object of interest.

The second filter relates to the tendency of object detector algorithms to maximize recall of returned boxes. For example, to avoid any missed detections, the selective search of [136] generates on average 1,000 to 3,000 candidate boxes per image, whereas a DPM detector of [41] visits more than 100,000 locations for a normal sized image, a number of visits that is feasible because of the dynamic programming involved. We compute a saliency map [56] of the image to discard the detections $D_i$ that do not occur in regions less likely to contain the actual object. The saliency score is helpful when the image is not cluttered with too many objects. Empirically,
local alignments for fine-grained categorization

Figure 29: Objectmap localization. The result of the GrabCut segmentation algorithm is shown in the first row, when a bounding box is provided by the user, a common methodology in the fine-grained literature, see [11, 26, 158]. The objectmaps computer with an object detectors, here selective search of [136], are shown in the second row. For these objectmaps no user input of any form is required. Naturally, having no bounding box usually results to a less accurate segmentation, especially when other salient objects appear in the image as well. However, objectmaps still tend to concentrate on the fine-grained object, usually including some additional background of course.

we have observed that this is often the case with certain fine-grained categories such as birds, as taking a picture of a fine-grained object, e.g., a bird, implies a special interest to the particular sub-category and often results in a clear photo of the object.

After having obtained the objectmap for the fine-grained object in the image, we proceed with the segmentation. The segmentation component of our approach is based on GrabCut, see [112]. GrabCut uses a gaussian mixture model, which groups pixels with similar appearance together, such that the foreground is separated from the background. The gaussian mixture model is trained iteratively in an alternate fashion. During the first step the foreground and background probability density functions are updated, based on the current pixel foreground/background labels. During the second step, the pixel labels are re-estimated via graph-cut inference, using the updated foreground and background probability density functions to calculate the unary terms and the image gradients for the binary terms.

Using objectmaps we end up with figure-ground segmentations, as shown in Figure 29. While the segmentation masks are not perfect, we recover sufficient spatial support for the object for most of the images.

5.3.2 Alignments and Partitions

Supervised Alignments

In a supervised setting the ground truth locations of basic object parts, such as the beak or the tail of birds, are available in the training set. This is a typical scenario when the number of images is limited, so that human experts can provide annotations at such a fine level of granularity. In the supervised alignment setting, we aim at accurately aligning the test image with a small number of training images. Then, we can use the common frame of reference to predict the part locations in the test image.

Different from general object categories that are often visually quite dissimilar from one another, fine-grained sub-categories typically share a great deal of similarities, mainly regarding...
their shape, their appearance and their poses. Hence, if the exterior shape of a fine-grained object is accurately captured, one can compare it with similar shapes in the training set and align the respective fine-grained objects. Note that, at this stage, it does not matter whether these are images that belong to the same sub-category or not. In order to acquire an impression of the shape of the object, we proceed with extracting a figure-ground segmentation mask $S_i$ of the fine-grained object in image $i$. The segmentation mask is usually not perfect: often background is included, foreground is omitted, or the mask delineates inner edges of the object, not representative of the exterior shape, see Figure 29. Furthermore, the interior of the shape mask carries little information regarding the pose of the object. We therefore suppress all the interior shape appearance by setting the inner pixels of the segmentation mask to zero.

After having extracted the segmentation mask, we encode the object shape by computing a HOG feature, that is $h_i = H(S_i)$. A HOG descriptor forms a high-dimensional space, which in theory may be populated by all shapes possible. Fine-grained objects, however, tend to have similar shapes and are seen in a limited repertoire of poses. More specifically, the observed exterior shapes reside on a lower dimensional manifold. Given an unseen fine-grained object, we can expect that its shape will probably be located in a specific region on this manifold. The fine-grained objects on this part of the manifold will have similar exterior shapes and, due to the anatomical constraints of the super-category they belong to, also similar poses on average. We take advantage of this principle to retrieve the $N$ training exemplar images $I_N$ from the training set $D_t$ which have the most similar exterior shapes using a query-by-example setting. In the end we have a shortlist of exemplar objects with similar poses, although no supervision was required regarding object poses or geometry. Examples of pose retrieval given an object of interest are shown in the upper row of Figure 30.

Figure 30: **Supervised alignment.** (a) Predicting part locations: in the top left, we have a query image, for which we want to predict part locations. On the right, we have the nearest neighbor training images, their HOG shape representations (top) and their ground truth part locations (bottom), based on which they were retrieved. Regressing the locations from the nearest neighbors to the test image we get the predicted parts, shown as the colorful symbols (bottom left). Although we rely on exterior shape only, the part locations can be found consistently. (b) Describing parts using all the information within a square patch (shown left) gives inferior results compared to using only the information within the square patch that falls inside the object’s segmentation mask (shown right).
Having retrieved the exemplar images with the most similar poses, we are in a position to transfer information from the training set to the test images. For the training exemplars $I_N$ we know the ground truth part locations $x$, as well as the appearance of the image regions that surround the parts $V_x$. In order to calculate the locations of the part of interest in the test image $I_q$, we employ a part pooling function $f(\cdot)$, that is
\[
\hat{x} = f(I_q; x_i, V_{x_i}), i \in I_N
\] (5.3)

The part pooling function $f$ can vary in sophistication. We can apply simple average pooling, or we can learn part appearance models in a similar manner to [8, 41]. In average pooling the predicted part locations are computed as the average of the respective part locations in the nearest neighbor images of the training set. This works well because the nearest neighbour images are well aligned to the query image. Note also that the appearance of the part in the nearest neighbour images is not used in this setting. We have experimentally witnessed that average part pooling yields accurate results, accurate enough to recover rough alignments. To ensure maximum compatibility we apply the above procedure for all the training and all the testing images in the dataset, thus acquiring predicted part locations for all the objects in the dataset.

**Partitioning supervised alignments.** We know the location of the part centers. Next, we need to define the shape of the parts, given these centers. We consider two strategies, that is *square partitions* and *segmented square partitions*, see Figure 30b.

Square partitions. The first strategy is related to most part-based models like [41]. Given the partition centers $\alpha$, we sample local descriptors every $d$ pixels from a square region $R_{sq} = \{(x,y)| \alpha_x - T/2 < x < \alpha_x + T/2, \alpha_y - T/2 < y < \alpha_y + T/2\}$. Square partitions capture both object and background appearance.

Segmented square partitions. The second strategy bears close resemblance to the first one, the difference being that we now take into account also the segmentation mask that gives a spatial support for the objects. For segmented parts we sample only in the common area between the designated part region and the segmentation mask, that is $R_{sg} = R_{sq} \cap S_i$. Segmented parts better capture the object of interest, at the expense of including less context, since descriptors are sampled only within the segmentation mask.

Of course, more strategies can be imagined for extracting partitions for supervised alignments. Scale invariance could be helpful for example. However, introducing scale invariance for partitions comes at the cost of increased complexity and is therefore not considered in the current work.

**Unsupervised alignments**

In contrast to the supervised case, in the unsupervised scenario we assume that no ground truth is provided regarding the part locations of the images in the training set. In the absence of such a ground truth, it does not make sense to align the test image to a small subset of training images. Instead, we derive a frame of reference based on the global object shape, inspired by local affine frames used for affine invariant keypoint description [93]. More specifically, given the location $x_s$ of the pixels on the segmentation mask $S$ we fit a 2-D ellipse, whose two axes are computed as
\[
a_j = \bar{x}_s + \hat{e}_j \sqrt{\lambda_j}
\] (5.4)
where $\lambda_j$ and $\hat{e}_j$ are the $j$-th eigenvalue and eigenvector of the covariance matrix $C = (x_s - \bar{x}_s)(x_s - \bar{x}_s)^T$ and $\bar{x}_s$ is the average location of the mask pixels. Ideally, the ellipse should follow
the “spine” of the object. We show examples of estimated poses and their local 2-d geometry in Figure 31a.

Since objects appear in a variety of poses, often placed in confusing backgrounds, the segmentation masks are usually not perfect. To minimize such negative influence, we use all the pixels of the foreground segmentation mask for fitting the ellipse.

**Partitioning unsupervised alignments.** For unsupervised alignments one does not have much certainty regarding the object pose. Hence, simple, yet consistent alignment geometries are required to robustly describe similar object locations in previously unseen images.

**Gravitational partitions.** Given an elliptical pose for the fine-grained object, we need to define a reasonable orientation. Following anatomical observations we first consider the longer axis to be the principal one. Having chosen the direction of the principal axis, we need to define the starting point. We follow the gravity vector assumption, see [9, 101], and adopt the highest end point of the principal axis as its origin to arrive at gravity vector alignments. All partitions are orthogonal to the principal axis of the fine-grained object. Since this principal axis is often similar to the “spine” of the object, each partition captures indirectly a specific anatomical part. For example in the case of four gravitational partitions on birds, we roughly capture the “head”, the “torso”, the “belly” and the “tail” of the bird.

**Pyramidal partitions.** Gravity vector alignments are supposed to follow the principal direction of the object’s pose. Often, however, objects are photographed in a wild variety of poses, in which case gravitational alignments might return less consistent results. In this case, and since spatial pyramids have shown excellent result in image-level classification, see [72], one can compute pyramidal partitions centered in the centre of gravity for the estimated elliptical pose. Given an accurate local frame of reference, the pyramidal partitions capture in their quadrants semantically meaningful regions of the fine-grained object. Furthermore, by vertically mirroring the training images we inject invariance regarding the pose and directionality of the fine-grained object regions. For example, the upper quadrants capture the appearance of the head, while lower quadrants encode the appearance of the belly and the tail, no matter where the object is facing to. Our strategies for aligning unsupervised, gravitational or pyramidal, partitions are visually summarized in Figure 31b.

In theory, extracting unsupervised alignments is less accurate than extracting supervised ones. However, given an accurate spatial support provided by the obtained local frame of reference, and a robust set of rules for defining the pose of the fine-grained objects in different images, we are still able to obtain robust and consistent alignments over the entire database. Another advantage of such unsupervised alignments and their partitions is that they are consistently found in all the images of the whole dataset and not just a small number of them at a time. This contrasts to part detection methods like that of [41, 157], which require several part templates to ensure high precision. Since such templates are normally activated only for a portion of the training set, the number of available training data for learning the part appearance is effectively reduced.

**5.3.3 Description**

Our alignments, supervised or unsupervised, are designed to be rough. Thus, comparing corresponding regions of objects from different images is bound to be a noisy procedure. Relying on features that are designed to return precise representations, but also sensitive to common image transformations, such as HOG, [34], are likely to be suboptimal. This is a problem, which orderless descriptors, such as Fisher vectors, [104], do not face, as by design they do not encode
local alignments for fine-grained categorization

**Figure 31: Unsupervised alignments.** Random birds and dogs, after their shape has been recovered, see in (a) the black contour around the objects. Based on the geometry of the shape we estimate the pose of the object, assuming an elliptical form. Following the gravity vector assumption [101] of the green arrows, we obtain the dominant pose orientation, see red arrows. Different strategies for aligning unsupervised partitions in (b). In the top image we have gravitational alignments, that adopt an upwards dominant orientation after the gravity vector assumption. In the bottom image we have pyramidal alignments, centered according to the center of the elliptical pose.

any spatial properties of the appearance information. Nonetheless, in a fine-grained categorization setting describing localities is important. To inject such spatial awareness to orderless descriptors, we extract Fisher vectors from the well aligned, and therefore spatially constrained, partitions. By doing so we maintain a good amount of the spatial extent of the appearance, while avoiding being overly vulnerable to occasions where feature matching is challenging.

Fisher vectors are composed of the derivatives of the Fisher kernels with respect to the parameters of the codebook model used. For a gaussian mixture codebook model, with average terms $\mu_k$ and variances $\sigma_k$, the Fisher vector representation is $\phi = [\frac{\partial x}{\partial \mu_k}, \frac{\partial x}{\partial \sigma_k}]^T$. Due to the generally small number of words that Fisher codebooks use, unnormalized Fisher vectors are characterized by an over-burstiness of certain visual words. Therefore, for optimal performance, Fisher vectors are 

- first, power-normalized so that the large Fisher vector values become less accentuated, then 
- $\ell_2$-normalized, see [104].

These two subsequent normalizations can be viewed as a single, serial transformation $u$, that is $\hat{\phi} = u(\phi)$. Inspired by the findings of [104], we extend the above normalization procedure and propose applying recursive transformations $u$ to arrive at the final descriptor, which we will refer to as serial normalization. Namely, our final descriptor is of the form

$$\hat{\phi}_t = u(\hat{\phi}_{t-1}), \hat{\phi}_0 = \phi \quad (5.5)$$

where $t = 1, \ldots, T$ is the length of the recursion.

The intensity SIFT descriptors are extracted after converting images to the grayscale, thus discarding any color present in the image. Although color histograms are a straightforward way to add color in the image, it has been shown by [63] that extracting SIFT descriptors from different color channels of the original image makes better use of the color. We, therefore, extract SIFT descriptors from each channel of various color spaces. To be precise we extract RGB-, Opponent- and C-SIFT descriptors. Given that the fine-grained details that differentiate similar categories are often related to color, we expect these three color spaces to cover adequately the color variations present in various fine-grained sub-categories.
5.4 Experiments

5.4.1 Datasets

Animal categories and their sub-categories provide a challenging testbed for fine-grained categorization, as their taxonomy is usually connected to specific visual appearances. We evaluate our proposed methods on popular fine-grained datasets for recognition of bird species and dog breeds. As detailed next, these datasets capture different aspects of fine-grained categorization and we consider them complementary.

Birds

The Caltech-UCSD Birds-200-2011 dataset introduced by [148], is one of the most extensive ones in the fine-grained literature. The Birds dataset is composed of 200 sub-species of birds, several of whom bear tremendous similarities, see Figure 28a. The bird images in this dataset are distinguished only on a fine-grained level, since several of the sub-species belong to the same family. A characteristic example are the Forster’s Tern and the Least Tern sub-species in the far right of Figure 28a. As one circumscription reads for the Forster’s Tern for example, “the comma-shaped black ear patch in winter plumage is distinctive, but some other plumages are very confusing.”. Recognizing, therefore, such nuances is the key for their recognition. For each of the classes in the Birds dataset there are 30 training images and 30 testing images. We use the standard training/test split provided by the authors of [148]. In our experiments we use the ground truth part locations only during learning, unless stated otherwise. Furthermore, we use the ground truth segmentations, only for evaluation and not for any kind of learning.

Dogs

The Stanford Dogs dataset by [64] contains images from 120 different breeds. The dogs are visually easier to distinguish than birds, as only few breeds belong to a common, larger family. See for example how different the Norwich Terrier and the Scotch Terrier are in the right of Figure 28b. Dogs, however, are difficult to categorize for other reasons. Since, they are domestic animals, they are photographed in a great variety of poses, scales, viewpoints and often with other objects occluding them. Hence, for the fine-grained categorization of Dogs, before anything else, one needs first to recover poses accurately. In the Dogs dataset there are in total 12,000 annotated images provided for training and 8,580 images for testing. We use the standard training/test split provided by [64].

5.4.2 Technical details

Following common practice in the fine-grained literature [157–159] we mirror the training images in the datasets to double the size of the training set. We use the bounding boxes to normalize the size of the images, unless stated otherwise. Furthermore, we do not downscale the image like in [157, 158], as we found this has a severe impact on the accuracy. For example downsampling images with the maximum dimension being 250 pixels drops accuracy by 23% for Birds. Last, we note that only for the Birds dataset there exist ground truth part locations as well as ground truth segmentations. Therefore, for the experiments where such ground truth information is needed, whether for evaluation or learning, we report results on the Birds dataset only.

http://www.allaboutbirds.org/guide/forsters_tern/id
Figure 32: A fine-grained category-by-category comparison using parts encoded by Fisher vectors or by HOG. We report results on the 200 Birds sub-categories measured in terms of accuracy. Fisher vectors perform consistently better on parts than HOG, having an average accuracy of 52.5% versus 31.8%.

We extract SIFT descriptors using the VLFeat library [142]. We sample densely every 3 pixels and at multiple scales ([16x16], [24x24], [32x32], [40x40]). After extracting the SIFT descriptors we reduce their dimensionality to 64 by PCA. To arrive at Fisher vectors we use a Gaussian mixture model with 256 components and use both the derivatives with respect to $\mu$ and $\sigma$, for a total of 32,768 dimensions. For the Fisher vectors we evaluate serial normalizations for a varying number of recursions, as described in Section 5.3.3. For HOG features we use the VLFeat implementation on a standard spatial grid of 8 pixels width per tile and then $\ell_2$ normalize them. Unless stated otherwise, we apply the standard normalizations per feature type, that is power and $\ell_2$ normalization for Fisher vectors and $\ell_2$ normalization for HOG. Finally, as a classifier we use the linear SVM PEGASOS implementation [118] with a fixed parameter $C = 10$.

We use the standard evaluation metric for these datasets, that is the category normalized mean accuracy over all the sub-categories within a dataset. Accuracy is defined as the number of correctly classified pictures for a certain sub-category, divided by the total number of pictures of that sub-category. All results are reported strictly on the test sets.

5.4.3 Experiment 1: What descriptors?

Setup. In this first experiment we evaluate whether rigid descriptors, such as HOG [34], or distribution-based descriptors such as Fisher vectors [104] are more accurate for describing parts in fine-grained categorization. To ensure a fair comparison, as well as to test the maximum recognition capacity of parts for such a task, we use the ground truth part locations for both the training and test sets. We also investigate different parameterizations for Fisher vectors. We experiment on the Birds dataset using the provided part annotations. To minimize redundancy...
Occasionally, encoding parts by HOG is better than Fisher vectors. The Rhinoceros Auklet birds in the first row have a very characteristic white horn on their beaks and two elongated white feather brows next to their eyes and their beaks. The shape-sensitive HOG better captures the appearance of those birds. Similarly, the Brandt Cormorant species also has a very distinctive sigmoid shape, also better described with HOG. In the majority of cases, however, Fisher vectors are significantly more accurate, see Figure 32.

due to the overlap, we use the following seven parts only, which together cover the complete silhouette of a bird: beak, belly, forehead, left wing, right wing, tail and throat. Both Fisher vectors and HOG are extracted on 100x100 pixel windows. Fisher vectors and HOG are also extracted from the whole bounding box. In the end we concatenate the Fisher vectors together into a single vector and the HOGs together into a single vector.

We also evaluate the effect of applying serial normalizations of eq. (5.5) to the final accuracy. To avoid irrelevant factors influencing the results, we conduct this experiment again under an oracle setting and use the ground truth segmentation masks provided for Birds to compute a single Fisher vector representation per fine-grained object.

**Results.** We show the results for the different parameterization of the Fisher vectors in Figure 34. For 128-PCA, we apply the PCA matrix, thus de-correlating only and not reducing the SIFT vectors. We observe that having more gaussian components and more dimensions after PCA has a positive impact on the accuracy. To control the final feature dimensionality, as well as to be compatible with the state-of-the-art, in the following we will make use of 256 gaussian components and 64 dimensions after PCA.

In Figure 32 we visualize the comparison between Fisher vectors and HOG. Clearly, Fisher vectors are better in describing parts for fine-grained categorization than rigid descriptors like HOG. Where HOG scores an accuracy of 31.8% on average, the Fisher vectors result in a final average score of 52.5%. The reason is that HOG descriptors require quite precise part detection, so that the gradients are representative of the appearance. Fisher vectors, however, aggregate the information from a larger area, adding more flexibility to the representation. Two notable exceptions, where HOG outperforms Fisher vector, are shown in Figure 33. In the majority of cases, however, Fisher vectors are clearly better for describing fine-grained subcategories than HOG, as Figure 32 reveals, outperforming for 184 out of the 200 bird categories. From now on we report results using Fisher vectors for describing the appearance of parts and alignments.
Regarding the serial normalization of eq. (5.5) on Fisher vectors, we observe that optimal results are obtained after two recursions, that is $T = 2$, improving recognition over the standard power normalization by an absolute 2-3%. This conclusion was also confirmed in subsequent non-oracle experiments, improving recognition even up to 4% for color based features. Comparing with other popular normalization schemes, serial normalization significantly outperforms $\ell_1$ and $\ell_2$ normalizations. We note here that serial normalization resembles the process of selecting the $\alpha$ value for the power normalization $\text{sign}(x)|x|^\alpha$, with the benefit of needing fewer rounds of parameter fine-tuning. From now on we report all results after a $T = 2$ serial normalization of the computed Fisher vectors.

5.4.4 Experiment 2: What partitioning?

Setup. In this experiment we evaluate various partitions for the description of fine-grained objects.

For the supervised alignments we follow the same setup as in the previous experiment, using the same seven parts plus a Fisher vector extracted from the whole bounding box. We predict the location of these parts in unseen images using the top-20 nearest neighbors. When the majority of the nearest neighbors does not have a certain part, it is marked as absent for the unseen image and the corresponding part of the Fisher vector is set to the zero vector. Also, we repeat the same experiment using only the predicted location of the beak.

For the unsupervised alignments no ground truth part annotation is required, so we evaluate on both Birds and Dogs. After extracting the principal axis of the object of interest, we split the segmentation mask into aligned partitions. For the object-level Fisher vector we use only the
Table 11: Experiment 2: What type of partitioning for Birds? Supervised alignments are more accurate than a spatial pyramid kernel and an alignment based on the beak of a bird only, while being rather close to the theoretical accuracy of the oracle parts that score 52.5%. When considering the segmentation masks for the description of the supervised alignments as in the right picture of Figure 30b, the accuracy improves even further.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised segmented square alignments</td>
<td>57.6%</td>
</tr>
<tr>
<td>Unsupervised gravital alignments</td>
<td>51.6%</td>
</tr>
<tr>
<td>Supervised square alignments</td>
<td>50.2%</td>
</tr>
<tr>
<td>Unsupervised pyramidal alignments</td>
<td>49.2%</td>
</tr>
<tr>
<td>Fisher vector from segmentation masks</td>
<td>42.6%</td>
</tr>
<tr>
<td>2x2 spatial pyramid</td>
<td>39.8%</td>
</tr>
<tr>
<td>Supervised alignment on beak</td>
<td>37.8%</td>
</tr>
<tr>
<td>Fisher vector from bounding box</td>
<td>32.1%</td>
</tr>
</tbody>
</table>

pixels within the segmentation mask and not the whole bounding box. We also examine what is the effect of a varying number of parts on the final accuracy.

Finally, we provide comparisons with state-of-the-art methods reported on the same datasets. For this purpose, we first evaluate the significance of color in fine-grained categorization. Apart from grayscale SIFT features, we additionally extract SIFT features from the RGB, Opponent and C-spaces [63].

Results. We show the results of this experiment for Birds in Table 11. When considering supervised square alignments, we obtain 50.2% accuracy, a large improvement over the 39.8% from the 2x2 spatial pyramid. Comparing the individual accuracy differences, the supervised alignments perform consistently better than spatial pyramids for 141 of the 200 classes (data not shown). The reason is that birds are well aligned, so the Fisher vectors computed on the respective parts capture the same nuances that differentiate sub-classes more consistently.

We measure the accuracy of the estimated part locations with respect to the ground truth locations. To cancel out the different bounding box geometries we normalize the part locations. After normalization the average location error is 12%.

Interestingly, when considering the supervised segmented square alignments using the GrabCut based segmentations the recognition accuracy improves further, reaching 57.6% and outperforming all other methods. This translates to a 7% gain as compared to supervised square alignments. We can therefore deduce that segmentation masks are helpful not only for describing whole objects, as they are normally used [26, 27], but also for the description of individual parts or regions of the fine-grained object of interest. With an exception of the work from [4], who use poselet-inspired region detectors, we are not aware of any works that researched the potential of segmented parts for recognition.

We focus now on the case when no ground truth of the part locations is provided, neither for training nor for testing. For unsupervised gravital alignments we reach an accuracy of 51.6%. Having fewer partitions leads to a lower accuracy (48.4% for two partitions), whereas too many alignments bring little extra benefit (51.7% for seven partitions). Extracting four partitions therefore suffices and we will use this number throughout the rest of the experiments where
we extract unsupervised alignments, unless stated otherwise. Comparing the supervised and unsupervised alignments when using their optimal settings, we show the differences in Figure 35. We observe that the supervised ones improve the accuracy especially for the classes where unsupervised alignments exhibit lower accuracy visible in the right part of the figure.

For the Dogs dataset we present the results in Table 12. The unsupervised pyramidal alignments outperform the unsupervised gravital alignments. The reason is that dogs are seen in a considerably larger variety of poses, scales and occlusions. In fact, as it is often the case that only the dog face is visible, any method that attempts to discover semantically meaningful parts becomes weaker, as also observed from [26]. Hence, for super-categories like Dogs, where the sub-categories are found in varying and peculiar poses, precise pose normalization should precede the extraction of fine-grained details.

We conclude that extracting localized alignments or parts matters in a fine-grained categorization setting. Furthermore, given their high accuracy, as well as their independence from ground truth part annotations, unsupervised alignments are appealing compared to supervised ones.

State-of-the-art comparison given bounding boxes. First, we evaluate the importance of color descriptors in fine-grained categorization tasks. In this experiment, we use the ground truth bounding boxes, as this is also done by the methods we are comparing against. The results after the addition of color are available in Figure 36a for Birds and in Figure 36b for Dogs. We observe that color consistently improves accuracy. From individual color channels only Opponent-SIFT performs well, increasing accuracy from 51.6% to 62.7% for Birds and from 45.2% to 51.5% for Dogs. When fusing the Fisher vectors computed on different color spaces with simple averaging, we reach an accuracy of 67.0% for birds and 57.0% for dogs. Hence, using multiple color...
Table 12: **Experiment 2: What type of partitioning for Dogs?** The unsupervised pyramidal alignments outperform the unsupervised gravital alignments. As also noted by [26], the reason is that dogs are seen in a considerably larger variety of poses, scales and occlusions.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsupervised pyramidal alignments</td>
<td>45.2%</td>
</tr>
<tr>
<td>Unsupervised gravital alignments</td>
<td>42.9%</td>
</tr>
<tr>
<td>2x2 spatial pyramid</td>
<td>42.8%</td>
</tr>
<tr>
<td>Fisher vector from segmentation mask</td>
<td>40.1%</td>
</tr>
<tr>
<td>Fisher vector from bounding box</td>
<td>36.2%</td>
</tr>
</tbody>
</table>

Figure 36: **Experiment 2: Adding color given bounding boxes.** (a) For Birds when considering the color information the accuracy becomes higher than the 51.6% obtained with grayscale only. More specifically, we obtain 60.0% by using C-SIFT, 61.5% by using RGB-SIFT and 62.7% by using Opponent-SIFT. When fusing the Fisher vectors computed on different color spaces with late fusion, the accuracies improve further to 67.0%. (b) For Dogs we make similar observations: 45.3% with C-SIFT, 48.3% with RGB-SIFT, 50.1% with Opponent-SIFT and 55.1% with average late fusion, as compared to 42.9% when only grayscale SIFT is used. Color is beneficial for fine-grained categorization.

channels brings a clear advantage over only grayscale information, as known for general object and scene detection [63]. In fact the experimental results reveal that a right use of color has an even stronger impact on the categorization of fine details, at least when animal species are considered.

Next, we compare state-of-the-art methods on fine-grained categorization, which also assume that the bounding box around the object is available at runtime. The results are available in Tables 13 and 14 for Birds and Dogs respectively. We observe that for birds unsupervised gravital alignments arrive at good recognition rates of 67.0% compared to the very recent state-of-the-art. The closest competitor, the deep learning approach of [37] combined with pose normalization from [165], reaches an accuracy of 65%. DeCAF makes use of large deep learning networks composed of 7 layers that require elaborate pre-training on many labeled images from 1,000 classes from ImageNet. Similar results are observed for Dogs, where unsupervised pyramidal alignments score 57.0% average accuracy. The closest competitor is the recent work of [26],
Table 13: Experiment 2: Comparison with state-of-the-art for Birds given bounding boxes. Unsupervised alignments outperform the state-of-the-art. Note here that the deep learning method of [37] makes use of extra labeled data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsupervised gravitational alignments</td>
<td>67.0%</td>
</tr>
<tr>
<td>Donahue et al. [37]+ Zhang et al. [165]</td>
<td>65.0%</td>
</tr>
<tr>
<td>Chai et al. [26]</td>
<td>59.4%</td>
</tr>
<tr>
<td>Donahue et al. [37]</td>
<td>58.8%</td>
</tr>
<tr>
<td>Berg et al. [11]</td>
<td>56.9%</td>
</tr>
<tr>
<td>Zhang et al. [165]</td>
<td>50.1%</td>
</tr>
<tr>
<td>Jia et al. [59]</td>
<td>38.9%</td>
</tr>
</tbody>
</table>

Table 14: Experiment 2: Comparison with state-of-the-art for Dogs given bounding boxes. Unsupervised alignments outperform the state-of-the-art.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsupervised pyramidal alignments</td>
<td>57.0%</td>
</tr>
<tr>
<td>Chai et al. [26]</td>
<td>45.6%</td>
</tr>
<tr>
<td>Yang et al. [157]</td>
<td>38.0%</td>
</tr>
<tr>
<td>Bo et al. [14]</td>
<td>36.0%</td>
</tr>
<tr>
<td>Khosla et al. [64]</td>
<td>22.0%</td>
</tr>
</tbody>
</table>

reporting an accuracy of 45.6%. We conclude that unsupervised alignments achieve state-of-the-art recognition rates for fine-grained categorization.

5.4.5 Experiment 3: Automatic fine-grained categorization

Having the bounding box location is a useful piece of information, as it separates, albeit roughly, the object of interest from the majority of the background. However, in most realistic scenarios bounding boxes are not available. In this experiment we examine the effectiveness of fully automatic fine-grained categorization, a process that entails automatic detection, segmentation and categorization of the fine-grained objects. To this end we first evaluate the importance of accurate segmentation in an oracle setting, by simulating added noise on ground truth segmentation masks. Then, we evaluate automatically detecting, segmenting and categorizing fine-grained objects.

Experiment 3A: Segmentation accuracy

Setup. In this experiment we evaluate the significance of accurate segmentations in a theoretical fine-grained categorization setting, where we assume that perfect segmentations for all fine-grained objects are available. We perform this experiment on the Birds dataset, as it is the only one for which ground truth segmentation masks are available. To make sure that conclusions reflect only the importance of segmentation accuracy, we extract a single Fisher vector from within the segmentation mask area, without considering any kind of partitionings. We start from the perfect ground truth segmentations, then generate artificially foreground or background noise.
Figure 37: Experiment 3A: The effect of segmentation accuracy in fine-grained categorization oracle segmentations on Birds. Noisy segmentation masks always hurt accuracy. However, missing superpixels of the ideal object segmentation is noticeably more harmful than including excessive background.

To generate the artificial noise we first oversegment the image into superpixels using [42]. Then, for the background noise we include extra superpixels neighboring the perfect segmentation mask, while for the foreground noise we exclude superpixels from the foreground mask. The superpixels are chosen such that the desired level of artificial noise is reached.

**Results.** We plot the results of this experiment in Figure 37. Inaccurate foreground segmentations appear to be quite harmful, see the left part of Figure 37. Foreground noise equals to missing foreground pixels. Losing a little bit of foreground, up to -20% has little impact on accuracy. However, when more foreground information is missing, the accuracy drops rapidly. When focusing on the right part of Figure 37, where background noise is added, we observe that the effect of imperfect segmentations is noticeable, but not dramatic. Indeed, adding 100% background noise, that is an area equal to the size of the bird, decreases the accuracy from 49.9% to 40.6%. If we expect the segmentation to be imperfect, either because of the low imaging quality or the challenging viewing conditions, a bias in favor of adding background than omitting foreground should be preferred.

**Experiment 3B: Fine-grained categorization without human intervention**

**Setup.** In this experiment we make no assumptions regarding the location of the object and want to compute a probability map, that encodes how likely is an object to be present at a particular image region. The first candidate is objectness [2], which was designed particularly for this purpose. We use the objectness parameters suggested in the latest release software, version 2.0, by the authors. For the objectmaps we use three state-of-the-art object proposal algorithms. Firstly, we use the deformable part model [41]. We lower the DPM detection threshold to -1.0, decided after visual inspection, to increase the number of detections returned. Secondly, we use
Table 15: Experiment 3B: Fine-grained categorization without human intervention. For birds unsupervised bounding box proposals [136] suffice for computing an accurate location for the object of interest. For dogs, however, where often multiple objects appear in the image, supervised bounding box proposals, [41], are more accurate.

<table>
<thead>
<tr>
<th>Alignments</th>
<th>Objectness</th>
<th>DPM</th>
<th>Selective search</th>
<th>Prime proposals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birds</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unsupervised gravital</td>
<td>32.7%</td>
<td>36.6%</td>
<td>40.6%</td>
<td>39.8%</td>
</tr>
<tr>
<td>Unsupervised pyramidal</td>
<td>31.7%</td>
<td>33.4%</td>
<td>38.6%</td>
<td>40.8%</td>
</tr>
<tr>
<td>Dogs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unsupervised gravital</td>
<td>29.4%</td>
<td>36.8%</td>
<td>30.4%</td>
<td>30.0%</td>
</tr>
<tr>
<td>Unsupervised pyramidal</td>
<td>31.4%</td>
<td>36.8%</td>
<td>34.0%</td>
<td>32.6%</td>
</tr>
</tbody>
</table>

Table 16: Experiment 3B: Comparison for Birds with state-of-the-art, without human intervention. Late fusion of unsupervised gravital alignments increases accuracy significantly.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late fusion</td>
<td>53.6%</td>
</tr>
<tr>
<td>Opponent-SIFT</td>
<td>51.6%</td>
</tr>
<tr>
<td>RGB-SIFT</td>
<td>49.0%</td>
</tr>
<tr>
<td>C-SIFT</td>
<td>48.9%</td>
</tr>
<tr>
<td>Grayscale</td>
<td>40.6%</td>
</tr>
<tr>
<td>[164]</td>
<td>28.2%</td>
</tr>
</tbody>
</table>

selective search [136] to generate object proposals. Last, we use the recently proposed prime proposals [86]. As objectness, DPM, selective search and prime objectmaps serve the same purpose, for clarity we will refer to all of them as objectmaps during the evaluation. We include comparisons with state-of-the-art methods that also do not require a location for the fine-grained object at runtime.

**Results.** We present the results for the Birds dataset in the first two rows of Table 15. The highest accuracy is obtained using the selective search and the prime objectmaps with unsupervised gravital and pyramidal alignments respectively. Their accuracy in the range of [40.6-40.8]% is a competitive result, when compared to the upper bound of 51.6% obtained where the bounding box locations are available, see unsupervised alignments on grayscale SIFT in Table 11. As in the previous experiment, we also consider the addition of three color spaces for the selective search objectmaps, see Figure 16. The results are consistent with the conclusions of the previous experiment. Extracting Fisher vectors from the Opponent-, RGB and C-SIFT spaces increases accuracy to 51.6%, 49.0% and 48.9% respectively. Applying late fusion using all color spaces as well as grayscale SIFT, we arrive at a final accuracy of 53.6%. For comparison, the automatic system from [164], that requires several part annotations during training, reports an accuracy of 28.2%. Note here that the selective search and prime objectmaps are fully unsupervised, requiring no human provided boxes, not even for training images, keeping the amount of human intervention to the minimum of providing only image-level annotations for the training set. The reason for their good performance in recovering bird locations is that birds often appear in isolation, with few other objects in the image. As a result, the selective search and prime bounding boxes usually concentrate around the most prominent object, which is a bird in most cases.
For the dogs dataset the results are shown in the last two rows of Table 15. For dogs, that often appear in a cluttered environment with many other objects, deformable part objectmaps work best, be it for gravital or pyramidal alignments, reaching an accuracy of 36.8% for both cases. After the addition of color on deformable part objectmaps, we obtain similar improvements as before, arriving at 47.2% and 49.0% for gravital and pyramidal alignments respectively.

We conclude that fully automatic fine-grained categorization is within reach. Using objectmaps as spatial priors allows unsupervised alignments to have a competitive accuracy, while requiring no user interaction regarding the parts nor the location of the fine-grained objects.

5.4.6 Qualitative analysis

Best recognized fine-grained objects. In Figure 38 we plot pictures from the Birds and Dogs categories for which unsupervised alignments reach the highest accuracy. The results for Birds are obtained with unsupervised gravital alignments, whereas for Dogs with unsupervised pyramidal alignments.

The fifteen birds with the highest recognition accuracy are characterized by an extensive color palette on their plumage. For example the European Goldfinch is easy to distinguish based on the intricate color patterns of red patches on their heads, followed by a black and white ring around their necks, their white belly, brown back and black and yellow wings. It appears that having several colors in different combinations and on different bird locations explains why these specific birds are easier to recognize than other species.

For Dogs we derive similar conclusions. First, as expected the different dog species have different colors, yet their chromatic palette is significantly more limited than for birds. Nevertheless, from the experimental results, see Figure 36, we know that color is also an asset. We conjecture that this is because for dogs the color gradients are more important than the color itself. The reason is that the color gradients locally reveal a particular type of texture, usually characteristic of the dog’s type of fur. For example the long, thin, “rasta”-like hair colored with different gradients of gray identify a Komondor, whereas the different gradients of brown and yellow identify the shiny fur of a Sussex spaniel. Hence, for Dogs extracting gradient based SIFT descriptors from different color spaces appears to be a good design choice as well, although the color variety is not as exotic as for Birds.
Figure 38: Experiment 4: Some of the best classified categories for unsupervised alignments for Birds and Dogs. For completeness we draw the detected boundaries after segmentation, see black contours. We observe that birds and dogs in these sub-categories have consistent appearance. It is noteworthy, especially for Birds, that most sub-species have very distinctive color patterns, which are well described by the color Fisher vectors we extract.
5.4 Experiments

Figure 39: Experiment 4: Two of the most confused pairs of bird categories, when only grayscale information is used. On the left we have the Forster’s Tern and Least Tern species, while on the right we have the Pelagic Cormorant and the Red faced Cormorant. The visual similarities between classes are remarkable, especially when no color is considered. Color is often necessary for telling such sub-categories apart.

What are the limits of visual features? Here we examine the other extreme, namely the categories which were difficult to recognize. In Figure 39 we show images of the two most confused pairs of bird categories, when only grayscale information is used: Forster’s Tern versus Least Tern and Pelagic Cormorant versus Red faced Cormorant. We observe that all the confused pairs belong to the same family of species. Indeed, their main differences are some colored details, e.g., the color of the beak. This is illustrated by a one-vs-one comparison of the birds in Figure 39 and the color versions of them in Figure 28.

Now, we turn our attention to the case when also color is considered. In Figure 40 we show images of two highly confused categories, when Opponent SIFT color features are considered: Great Grey Shrike versus Loggerhead Shrike and Caspian Tern versus Elegant Tern. These categories look very similar. It is likely that these birds are taxonomized based on some physiological, rather than purely visual, characteristics. Indeed, when looking up the taxonomical motivation for the Loggerhead Shrike and the Great Grey Shrike, we found that their main two differences are anatomical and geographical. First, for the Loggerhead Shrike the proportion between the head and the beak is usually larger. Second, the two species are parapatric. The Great Grey Shrike appears in Northern Eurasia and America, whereas the Loggerhead Shrike lives in the southern Mediterranean zone. This type of anatomical or geographical information is unlikely to be recovered from single pictures, where the birds appear in all sorts of angles, viewpoints and scales and the context is limited. We conclude that when this is the level of recognition required, expert knowledge, metadata, or perhaps analysis of the environment, as [35] would argue, might be necessary for guiding the machine further. For example, recognizing the Great Grey Shrike from the Loggerhead Shrike we could perhaps examine, whether the surroundings correspond to a subarctic or a temperate habitat respectively.

What makes a Bobolink a Bobolink? Here we exploit the properties of the linear SVM classifier, more specifically the additivity of the classification scores per feature dimension [46, 83]. Given a sub-category $c$ and its classification model $w^c$, we retrieve the dimensions $d$ with the largest, positive weight values $d = \arg_{d'} \max w^c_{d'}$, since they contribute the most to the final classification score. We then identify those pixels that have the strongest Fisher response for the dimensions $d$ of the sub-category classifier $w^c_d$. Due to monotonicity, the power and $\ell_2$ normalization do not influence the outcome of this qualitative evaluation. We visualize in Figure 41 results for the top

http://www.allaboutbirds.org/guide/loggerhead_shrike/id
local alignments for fine-grained categorization

Figure 40: **Experiment 4: Two of the most confused pairs of bird categories after adding color with Opponent SIFT.** The first pair of confused birds contains the Great Grey Shrike and Loggerhead Shrike species, whereas the second one the Caspian Tern and the Elegant Tern species. These birds species seem very similar to each other, even after the addition of color. It is likely that they are taxonomized based also on non-visual criteria, such as anatomical or geographical ones. Indeed, the main two differences between the Great Grey Shrike and Loggerhead Shrike are (a) the proportion between their head and their beak and (b) their habitat, with Great Grey Shrike living in the north and Loggerhead Shrike in the south.

20 dimensions ($|d| = 20$) for the 20 pixels with the strongest Fisher response using unsupervised alignments and Opponent SIFT.

Given the rough nature of the alignments we make several observations from the visualizations. First, it appears that the distinctive details appear consistently on similar locations on the fine-grained objects. For the *Boat tailed Grackle* the wide, round tail is the most distinctive detail. For the *Red face Cormorand*, it is the red patch on the bird’s head. An interesting case is the *Hooded Marganser*. What is considered very distinctive for this bird are the bright yellow eyes and secondarily the black and white stripes on its breast. As most birds have dark eyes, a brightly colored eye makes the difference. On the contrary, the large back of the head is not considered very discriminative and would probably be better captured by HOG. Overall, it appears that Fisher operates as a spatial hashing function, that builds a correspondence between spatial details and certain feature dimensions. As a result, although a more precise object or part localization is always welcome, employing features, such as Fisher vectors, may largely have the similar effect.

Furthermore, we generally observe that the most prominent information lies usually on the head. Placing special importance on detecting the head is therefore justified and may bring significant accuracy benefits, as has also been shown by [26, 80, 100]. Finally, we answer that a Bobolink is made by angular beaks and very sharp, black and yellow edges around the head and the neck of a bird.

5.5 conclusions

We aim in this chapter for fine-grained categorization without human interaction. Different from prior work, we show that localizing distinctive details by roughly aligning the object of interest allows for successful categorization of fine-grained sub-classes. In cases when an object pose can be confidently extracted, it is beneficial to focus first on recovering the pose and then detecting the interesting part locations: the anatomical constraints imposed by a detected pose make sure that the parts do not drift away.

We perform experiments on the challenging CUB-2011 dataset composed of 200 bird species and on the Stanford Dogs datasets composed of 120 dog breeds. Under a controlled, oracle setting the experimental results indicate that for rough alignments, distribution based features, such as Fisher vectors, are a better choice than rigid features, like HOG (Fig. 32).
Figure 41: What makes a Bobolink a Bobolink? Visualizing why birds are recognized as certain sub-species. It appears that the distinctive details appear consistently on similar locations on the fine-grained objects. Furthermore, we generally observe that the most prominent appearance detail lies usually on the head.
We furthermore proceed with performing fine-grained categorization on unseen images, obtaining high recognition rates (Table 11, 12). What is more, the experiments reveal the importance of color SIFT in the recognition of fine-grained sub-species (Tables 13, 14), arriving at state-of-the-art results even when compared with deep learning approaches that make use of extra data. We attribute the superiority of local alignments encoded with color Fisher vectors to two factors: first, the rough, but consistent grouping of spatially neighboring fine-details and second, the potential of the Fisher vectors in describing such fine-details, even when the latter are not precisely localized.

We establish the importance of segmentation for fine-grained categorization. Naturally, a better segmentation results in a better accuracy. In the presence of segmentation noise, it better be background noise (Fig. 37). In absence of perfect segmentation solutions, we conclude that algorithms favoring the inclusion of background instead of the exclusion of foreground should be preferred for fine-grained categorization.

Guided by the conclusions from segmentation, we attempt to recover the spatial support of a fine-grained object, even in the absence of a user-provided bounding box. The extracted objectmaps, built on off-the-shelf object hypothesis algorithms, provide a good enough spatial support for the fine-grained object of interest. For fully automatic fine-grained bird categorization we obtain an accuracy of 53.6%, where the previous best was 28.2% reported by [164] (Table 16).

Finally, our qualitative analysis reveals that Fisher operates as a spatial hashing function, that builds a correspondence between spatial details and certain feature dimensions (Fig. 41). Therefore, even though a more precise object or part localization is always welcome, employing features, such as Fisher vectors, may largely have a similar impact. We, furthermore, observe that computer vision alone cannot solve all categorizations, as the subtle species differences might be anatomical, epochal, or geographical (Fig. 40). In such situations, use of expert knowledge, active learning or metadata would be necessary. For the majority of cases, however, local alignments allow for accurate, and inexpensive, categorization of fine-grained categories.