Understanding Objects in Detail with Fine-grained Attributes

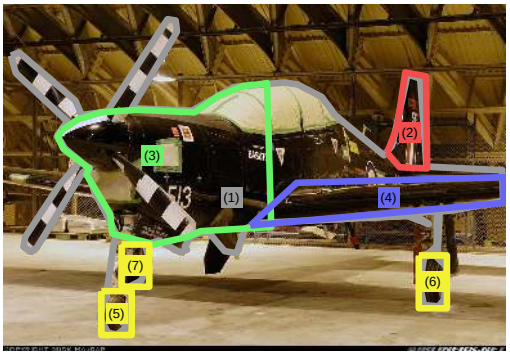
Andrea Vedaldi[1] Siddharth Mahendran[2] Stavros Tsogkas[3] Subhransu Maji[4]
Ross Girshick[5] Juho Kannala[6] Esa Rahtu[6] Iasonas Kokkinos[3]
Matthew B. Blaschko[3] David Weiss[7] Ben Taskar[8] Karen Simonyan[1]
Naomi Saphra[2] Sammy Mohamed[9]

Abstract

We study the problem of understanding objects in detail, intended as recognizing a wide array of fine-grained object attributes. To this end, we introduce a dataset of 7,413 airplanes annotated in detail with parts and their attributes, leveraging images donated by airplane spotters and crowdsourcing both the design and collection of the detailed annotations. We provide a number of insights that should help researchers interested in designing fine-grained datasets for other basic level categories. We show that the collected data can be used to study the relation between part detection and attribute prediction by diagnosing the performance of classifiers that pool information from different parts of an object. We note that the prediction of certain attributes can benefit substantially from accurate part detection. We also show that, differently from previous results in object detection, employing a large number of part templates can improve detection accuracy at the expenses of detection speed. We finally propose a coarse-to-fine approach to speed up detection through a hierarchical cascade algorithm.

1. Introduction

Image-based modeling is perhaps one of the most successful paradigms in image understanding. Image-based models capture objects as two-dimensional patterns, leveraging the power of statistical learning to characterize their variability and recognize them in images. The appearance of such patterns can be described by orderless statistics such as bag-of-visual-words or more sophisticated discriminative templates [16] accounting explicitly for object deformations, occlusions, and multiple aspects. While these models are relatively rich, comparatively little attention has been dedicated to the *detailed structure* of objects, particularly from a semantic viewpoint. For example, glancing at an image such as Fig. 2 we not only see a plane, but a "plane with two wings, retractable single-wheeler undercarriages under the wings, pointy nose with a four-blade, black-and-white, striped propellor, a small round cockpit window, etc." Current object models would fail to extract any of these properties, usually referred to as "attributes". In general, an *attribute* is any visual property that has a semantic connotation, such as the *redness* of an apple or the *roundness* of a nose. Attributes capture information beyond the standard phraseology of object categories, instances, and parts, and can significantly enrich object understanding. At the same time, attributes are often modeled as holistic properties of objects, disregarding their compositional and local nature. For example, a bird species could be characterized as having "short *wings*", a "dotted pattern around the *neck*", and an "orange *beak*". A face could be described as having



1 **airplane** facing-direction: SW; is-airliner: no; is-cargo-plane: no; is-glider: no; is-military-plane: yes; is-propellor-plane: yes; is-seaplane: no; plane-location: on ground/water; plane-size: medium plane; undercarriage-arrangement: one-front-two-back; wing-type: single wing plane; airline: UK–Air Force; model: Short S-312 Tucano T1
2 **vertical stabilizer** tail-has-engine: no-engine 3 **nose** has-engine-or-sensor: has-engine 4 **wing** wing-has-engine: no-engine 5 **undercarriage** cover-type: retractable; group-type: 1-wheel-1-axle; location: front-middle 6 **undercarriage** cover-type: retractable; group-type: 1-wheel-1-axle; location: back-left 7 **undercarriage** cover-type: retractable; group-type: 1-wheel-1-axle; location: back-right

Figure 1. **Beyond object detection: detailed descriptions.** An example annotated airplane in the proposed AirplanOID dataset. Our aim is to investigate models that understand object categories in detail, generating rich descriptions of each object instance.

[1]*University of Oxford;* [2]*Johns Hopkins University;* [3]*École Centrale Paris / INRIA-Saclay;* [4]*Toyota Research Institute Chicago;* [5]*University of California at Berkeley;* [6]*University of Oulu;* [7]*Google Research;* [8]*University of Washington;* [9]*Stony Brook University.*

Global attribures
facing-direction ∈ {N, NE, E, SE, S, SW, W, NW}
is-airliner, is-cargo-plane, is-glider,
is-military-plane, is-propellor-plane, is-sea-plane ∈
{true, false}
location ∈ {on-ground/water, landing/taking-off, in-air}
size ∈ {small, medium, large}
airline ∈ {AirFrance, Easyjet, AirCanada, ...}
model ∈ {Boeing747, ShortS-312Tucano084T1, ...}

Wings
type ∈ {single-wing, biplane, triplane}
has-engine ∈ {none, embedded, 1-on-top, 2-on-top, 3-on-top,
1-on-bottom, 2-on-bottom, 3-on-bottom}

Vertical stabiliser
tail-has-engine ∈
{none, 1-on-top, 2-on-sides, 3-on-top-and-sides}

Nose
has-propeller-or-sensor ∈ {none, propeller, sensor}

Undercarriage
undercarriage-arrangement ∈
{1-front-more-back, 1-back-more-front, other}
location ∈ {front-left, front-middle, front-right,
back-left, back-middle, back-right, }
group-type ∈ {1-wheel-1-axel, 2w1a, 4w2a, 6w3a, 14w7a}
cover-type ∈ {retractable, fixed-outside,
fixed-inside, fixed-outside-with-cover}.

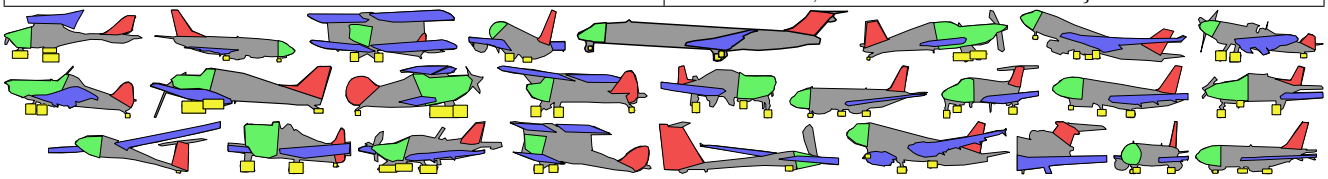


Figure 2. **AirplanOID data.** Each of 7,413 airplane instances is annotated with segmentations for five part types (bottom) and their modifiers (top). The data internal variability is significant, including modern large airliners, ancient biplanes and triplanes, jet planes, propellor planes, gliders, *etc*. For convenience, airplanes are divided into "typical" (planes with one wing, one fuselage, and one vertical stabilizer) and "atypical" (planes with a different structure); this subdivision can be used as "easy" and "hard" subsets of the data. Several detailed modifiers are associated to parts. For example, the undercarriage wheel group modifier specifies whether an undercarriage has one wheel on one axel, two wheels on one axel and so on.

a "round *nose*", "bulging *eyes*", "short *hair*", and "small *ears*". In all these examples attributes act as *modifiers of object parts*, with a clear compositional structure which is often disregarded in attribute modeling, partially due to the lack of suitably annotated data that could support the development of such models.

In this paper, we address this gap and look at the problem of understanding **Objects in Detail** (OID), intended as describing an object and its parts with a rich set of semantic attributes. In particular, we investigate how parts and their attributes can be modeled and recognized in images and how detailed supervision about them can be used to train better object models and analyze them.

The first challenge is to find which parts and attributes are useful in describing objects. While resources such as WordNet can be used to extract a taxonomy of object categories, this is harder to do for generic object properties. We address this problem by using a "comparison principle": a part/attribute is informative if it can be used to *discriminate* similar objects by pinpointing *meaningful and specific differences* between them. We measure this empirically from experiments conducted on Amazon Mechanical Turk (AMT) where annotators are asked to *differentiate* pairs of objects. The results are then filtered semi-automatically by a statistical analysis and used to bootstrap a vocabulary of parts and attributes. We apply this principle to the class "airplane" (the first class of PASCAL VOC, Sect. 2), collecting and annotating and "OID dataset" of 7,413 images of airplanes spanning a hundred years of aviation, with segmented parts and attributes carefully annotated for each airplane part instance (Fig. 2). Compared to existing object datasets, by focusing on a single object category we can afford to collect *significantly deeper annotations* and use them as the basis for a type of analysis that is not supported by existing datasets. The dataset is publicly available at http://www.robots.ox.ac.uk/~vgg/data/oid/.[1]

Our second contribution is an analysis of the relation between parts and fine-grained attributes. Sect. 3 uses the detailed annotations in the data to analyze which parts or part combinations are more informative for the performance of an attribute classifier. One conclusion in Sect. 3.1 is that contextual information is a very strong predictor of local attributes. Another one is that accurate part detection is highly beneficial in the prediction of certain attributes.

Sect. 3.2 also shows that, contrary to what has been observed in other object detection tasks in the literature [45], detection of object parts can be improved by adding a significant number of part templates to a state-of-the-art detector. Hence our final contribution is a coarse-to-fine algorithm to efficiently and accurately detect parts by organizing multiple templates in a tree hierarchy. Our method achieves a 4- to 5-fold speedup without sacrificing accuracy.

1.1. Related work

Applications of visual attributes. Due to their semantic connotation, visual attributes are very powerful in human-centric applications such as *generating image descriptions automatically* [13, 14, 22, 30] or *searching images based*

[1] We already introduced a superset of these aircraft images for FGcomp 2013 [28], but without detailed annotations.

on detailed descriptions of their content [14, 23, 38]. The method of[23] uses *simile* attributes to characterise face instances (*e.g*. "a nose like Harrison Ford's") and [34] extend this to *relative properties* (*e.g*. "more masculine than Clive Owen", "less natural than a forest"). The method of [21] uses comparative (*e.g*. "more shiny than") and simile attributes for interactive query refinement. Attributes can also be used to *transfer expert knowledge to a computer vision system* in zero-shot learning [25, 43], in finding relations between object categories [34, 42], in incorporating *annotator's rationales* and other forms of feedback [8, 35].

Mining and annotating visual attributes. The selection of useful attributes is often left to the intuition of the researcher [13, 36]. In specific domains attributes may be extracted from field guides [26, 43] or documents downloaded from the Internet [3, 32]. The method of [33] starts from randomly generated classifiers and uses humans in the loop to find which ones may correspond to meaningful image attributes. [6, 27] use a comparison principle to corwdsource attributes from image collections.

Modelling and recognising visual attributes. While attributes may be conceptually attractive, they are useful only if they can be detected reliably in images. The work on modelling attributes is relatively limited, as most authors use off-the-shelf methods such as bag-of-visual-words [8, 9, 13, 14, 21, 25, 25, 30, 33–36, 39, 42–44]. Despite the fact that most object attributes are local (*e.g*. properties of parts), only a few authors account for locality explicitly [12, 22–24, 38, 41, 43]. Very few works consider the correlation between attribute occurrences [38, 44].

Attribute datasets. There are only a few datasets annotated with attributes, and fewer still with object attributes. Examples of the latter are the a-Yahoo/a-PASCAL datasets of [13], but these contain image-level annotations. CORE [12] contains coarse part and attribute annotations for several object categories, while our focus is the fine-grained description, which led us to obtain detailed annotations of a large number of images of one class. This trade-off is motivated by the necessity of obtaining a statistically acceptable sampling of subtle object variations. CUB-200 [40] also contains images of one object category annotated in detail (bird), but their attributes are specialized to the identification of bird species, while we are interested in general object properties.

2. Detailed object annotation methodology

Similar to standard tasks such as image categorization and object detection, describing objects in detail requires a suitable dataset. This section discusses the nuances involved in collecting a large set of detailed object annotations [10] and a methodology to do so efficiently.

As a running example we consider the class *airplane*, contained in standard datasets such as Caltech-101 [15], PASCAL VOC [11], and CORE [12]. The airplane class was selected because airplanes are largely non-deformable, simplifying object and part detection, but contain significant structural variably (biplanes, fighter jets, private jets, propeller planes, airliners, *etc*.), supporting a rich set of modifiers. The resulting annotated dataset, dubbed AirplanOID, comprises 7,413 images of airplanes with segmentations for five object part types as well as discrete labels for a number of modifiers, as listed in Fig. 2. Crucially, several modifiers apply directly to specific object parts (*e.g*. the number of wheels of an undercarriage), allowing to investigate the interaction between local and global semantic and modeling.

The next paragraphs illustrate how AirplanOID was collected, emphasizing insights of general applicability. The process is broken down into three phases: (i) collecting images, (ii) crowdsourcing attributes and parts, and (iii) crowdsourcing annotations.

(i) Collecting images. Rather than following the standard approach of drawing images from sources such as Google, *it was found that extracting images from specialized collections was much more efficient*. For airplanes, we downloaded 7,413 images from airplane spotters (http://www.airliners.net/); interestingly, several of them agreed to let us use their copyrighted material for research purposes for free. Not only these images are curated, but they also come with a significant amount of very detailed metadata (*e.g*. airplane model) that would be difficult to obtain otherwise. Similar collections are available for other object categories as well.

(ii) Crowdsourcing attributes and parts. In order to find useful parts and attributes to describe objects, the natural approach is to look at the terms that humans use to describe them. However, when asked to describe an object directly [37], annotators did not usually produce detailed information; instead, *asking about the differences between objects pairs was found to be significantly more effective* [27], producing a substantial list of candidate parts and attributes.

(iii) Crowdsourcing annotations. The list of candidate parts and attributes from (ii) was pruned to meet our annotation budget, selecting five airplane parts (airplane, wing, undercarriage, vertical stabilizer, nose) and several corresponding modifiers (Fig. 2). While parts and attributes are intuitive by design, it was found that providing clear specifications and instructions to annotators substantially improved the annotation quality. In particular *instruction were iteratively refined by looking at early batches of annotations, adding illustrations of typical annotation errors and how to correct them*. Each of the 7,413 images was submitted to AMT for annotation, collecting attribute labels and

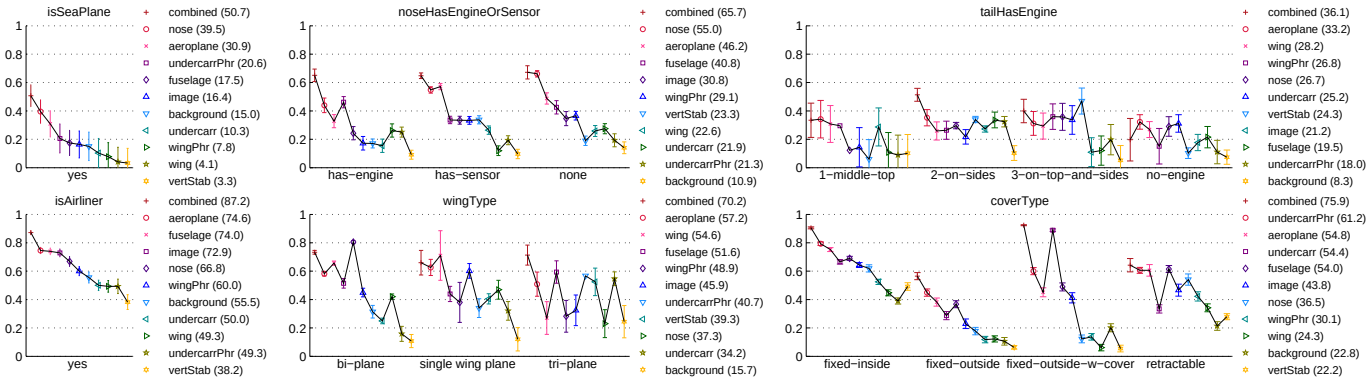


Figure 3. **Attribute prediction using local evidence**: local evidence, measured in correspondence of different object parts, is used to predict fine-grained attributes taking two, three, or four different values (left-to-right); the plots show the impact of different sources of local information on attributed prediction, quantified as Normalized AP (NAP). In each plot, parts are sorted by decreasing mean NAP.

Attribute	Method	Best	Second best	Combined	Airplane	Image
undercarriage cover type	g.t.	undercarriage phr. (61.2%)	undercarriage (54.4%)	75.9%	54.8%	43.8%
	detected	undercarriage phr. (37.3%)	nose (28.9%)	–	34.0%	–
tail has engine	g.t.	wing (28.2%)	wing phr. (26.8%)	36.1%	33.2%	21.2%
	detected	vertical stabiliz. (21.6%)	undercarriage phr. (19.6%)	–	20.1%	–
facing direction	g.t.	nose (49.0%)	fuselage (39.6%)	70.6%	52.0%	43.8%
	detected	nose (29.8%)	wing phr. (24.0%)	–	31.9%	–

Table 1. **Attribute prediction with ground truth and detected parts.** Only selected attributes shown due to space constraint.

part polygons (rectangles for *undercarriage*). The quality of the annotators was highly variable, but *maintaining a list of reliable annotators* addressed this issue successfully. For attribute labels, annotation noise was reduced by collecting from 5 to 10 redundant annotations (if after 10 rounds no label received a least 80% of the votes, the instance was declared ambiguous). Part polygons were also collected multiple times (from 3 to 6), but in this case verified by our team, as there is no easy way to average these annotations. To this end, a tool was developed to allow the team to work concurrently by selecting the best polygonal segmentation for each airplane/part instance, automatically ranking annotators and prioritizing the work of good ones. Overall, three weeks of intense work were sufficient to collect high-quality annotations for the 7,413 images. Nevertheless, part validation is still expensive (in our case it involved about ten researchers full time) and reducing this cost further is subject of current research.

3. The role of parts in detailed understanding

Our exploration of the OID problem starts by investigating the notion of part and its role in defining a compositional semantic of attributes. In particular, the first question is whether extracting parts is necessary or useful in predicting detailed object properties. While this may be a given, in fact we will see that the correlation between the general structure of the object and the local attributes make it possible to recognize certain attributes *without* localizing parts; however, this is not true for all of them. Given that detecting parts is important for at least a subset of the fine-grained tasks, the second question is how parts can be best modeled and detected. Here the surprising insight is that very detailed appearance models do not overfit the data but rather they surpass more regularized settings that have been found to be optimal for generic object detection.

3.1. The effect of parts on attribute prediction

A key benefit of the OID dataset (Sect. 2) is that it allows an in-depth analysis of attributes and their relation with object parts. We now use OID to (i) study which object parts are informative for which attributes and (ii) how part detection quality can impact attribute prediction performance.

To this end, we build attribute classifiers and combine them with ground-truth and detected parts. We investigate two standard models, Bag-of-Visual-Words (BoVW) and Deformable Part Models (DPM), using state-of-the-art implementations. The BoVW representation of an image window (corresponding to an entire image, an object, or one of its part) uses dense SIFT features at multiple scales, quantised into 2048 visual words, 1×1 and 2×2 spatial subdivisions, l^1 normalised histograms plus the square root feature map, as detailed in [5]. DPMs use the latest implementation available online (v5) [18]. The latter was modified to incorporate part-level supervision in some experiments.

Similar to [19], the OID data allows an analysis of the performance of a classifier and its failure modes in relation to properties and attributes of objects. However, while [19] is limited to diagnosing the detection problem, OID allows to move the study to the level of the individual parts and attributes. As in [19], we evaluate classifiers in terms of

Normalized Average Precision (NAP) in order to penalize inaccurate predictions while at the same time making different tasks comparable.

In the first experiment (Fig. 3), attribute classifiers are learned by restricting visual information to specific parts of the objects: the whole image, the whole airplane, the nose, the undercarriages, the wings, the vertical stabilizer, the fuselage (obtained by subtracting the other parts from the airplane body), undercarriage and wing phrases (a bounding box around the corresponding parts), and the background. For each part, a BoVW descriptor is obtained as explained above; part combinations obtained by stacking the corresponding descriptors are evaluated as well. Fig. 3 shows a subset of classifier performance comparison results.

The first observation (a) is that *in all cases the model using part combination outperforms any single part model*. The second observation (b) is that, while generally (but not always) attributes with a local semantic are best predicted by the corresponding part, *other non-overlapping parts* are excellent predictors as well (note that chance NAP is 1%). For example, the attribute `tail-has-engine` is predicted with mNAP (mean NAP) of 28.2% by the *wing phrase* part, and the *vertical stabilizer* is only the fifth best part predictor at NAP 24.3%. Observations (a) and (b) indicate that contextual information is complementary and often nearly as good (and sometimes better) than direct evidence. Thus, learning attributes from global object or image cues as commonly done [8, 9, 13, 14, 21, 25, 30, 33–36, 39, 42–44] may in fact pick up contextual evidence more than learning about the attribute per se.

In the second experiment (Tab. 1), the performance of attribute prediction is evaluated in function of the reliability of part detectors, repeating the experiment above, but using detected rather than ground-truth parts. The drop in performance when moving from ground truth to detected parts is in some cases substantial, suggesting that work focusing on part localization is likely to benefit significantly fine-grained object description.

3.2. Improving part detection

As noted in Sect. 3.1, part detection plays a major role in fine-grained attribute prediction. This section seeks ways of improving the detection of object parts, revisiting the question "Do we need more data or better models" for object detection posed by [45]. In their work they showed that on PASCAL VOC datasets a leading approach for detection, mixtures of deformable part-based models tends to saturate on performance with 3 mixtures on most categories. Even with the addition of 10× more data the authors noted that the performance of the models does not improve significantly — more mixtures simply added robustness by 'taking away' noisy training examples.

Do more mixtures help? We first perform an experiment

Part	$k = 6$	$k = 20$	$k = 40$	Shape
Nose	57	60	62	68
Vertical stabilizer	42	54	52	60
Wings (grouped)	15	19	22	28

Table 2. Part detection performance (MAP%) as a function of the number of components. Shapes results were obtained with $k = 40$ components, and part segmentations to initialize clusters.

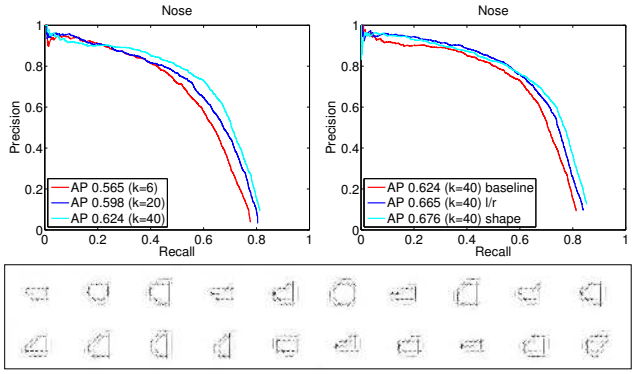


Figure 4. **Nose detection results.** Nose shape clusters learned by EM. Detection AP using $k = 6$, 20, and 40 mixture components based on aspect-ratio clustering to initialize the latent SVM (left). AP for the baseline clustering, left-right clustering from [18], and our supervised shape clustering for $k = 40$ (right).

where we train mixtures of root-only models by varying the number of mixture components $k = 6$, 20, and 40. We found that, for most parts, detection performance saturates at around 40 mixture components, which is an order of magnitude higher than the same number on PASCAL dataset. We use aspect-ratio based clustering to initialize the mixtures. Fig. 4 (top-left) shows that the performance of the learned models for detecting noses are respectively 57%, 60%, and 62%, improving consistently as the numbers of mixtures increase. Table 2 shows similar gains for other parts. This can be because unlike those in the PASCAL VOC dataset, our objects are of significantly higher resolution and variety, and hence can benefit from the details more mixture components can provide.

Do semantic attributes help? In addition to aspect ratio based clustering we can use a supervised left-right clustering which improves performance from 62% to 67% for the nose part (Fig. 4-right). Additionally, we use the segmentation of the noses to cluster the shapes using the HOG features of the foreground object. This initialization improves the performance to 68%. A similar trend is observed for other parts and the overall object as seen in the last column of Tab. 2. Thus better initialization using semantic attributes can improve detection accuracies a trend observed by several others [4, 29, 45].

4. Hierarchical part detection cascades

Having outlined the merit of using multiple shape clusters for part detection, we now address computational efficiency. The refined processing advocated in Sec. 3.2 incurs an increase in computational cost, as it requires the evaluation of multiple classifiers at every candidate part location.

One of our technical contributions is a method to efficiently process an image with a fine-grained model through a hierarchical part detection cascade. We build on the broader theme of sequential testing [1] and organizing multiple classifiers in a tree-structured hierarchy [2, 7], and integrate it with bounding-based detection [20].

We develop a coarse-to-fine algorithm that originally gets rough and quick score estimates for sets of similar components, and then recursively refines such scores by working with increasingly smaller sets of components. For this, we start in Sec. 4.1 by recovering an hierarchy of parts from a set of discriminatively trained components. Then, in Sec. 4.2 we use this hierarchy at test time to quickly recover filters that score above a predetermined threshold by recursively constructing probabilistic upper bounds on part scores lying below a tree node, and pruning accordingly.

4.1. Part hierarchy computation

We establish a tree-hierarchy to represent the k component filters learned in Sec. 3.2; we use agglomerative clustering and force the learned tree to be binary, ensuring that it has depth at most $\lceil \log_2 k \rceil$. As shown in Fig. 5, the leaf nodes of the hierarchy learned for the 'nose' part correspond to the individual components-filters while higher-level nodes represent the ensemble of filters below them.

Starting with the leaf nodes, we construct a $k \times k$ dissimilarity matrix D between parts, containing the alignment-based dissimilarity of components i and j:

$$D[i,j] = \min_{h',v'} \sum_{h,v} \left(f_i(h,v,d) - f_j(h+h', v+v', d)\right)^2 \quad (1)$$

where h, v, d are the horizontal, vertical, and direction indexes of a HOG template respectively, h', v' indicates an amount of translation applied to f_j, while we treat different sizes of f_i, f_j by zero-padding. We greedily pick the most similar pair, remove the respective rows and columns from D, and repeat until all leaves get paired. Each pair i, j is represented by its parent, l, in terms of the aligned mean:

$$f_l(v,h,d) = \frac{1}{2}\left(f_i(v,h,d) + f_j(v+v^*, h+h^*, d)\right), \quad (2)$$

where (v^*, h^*) is the minimizer of Eq. 1. We repeat this procedure at the next level, halving the number of nodes present at every hierarchy level; for $k = 2^i$, the hierarchy will thus contain $2k-1$ nodes and $i+1$ levels.

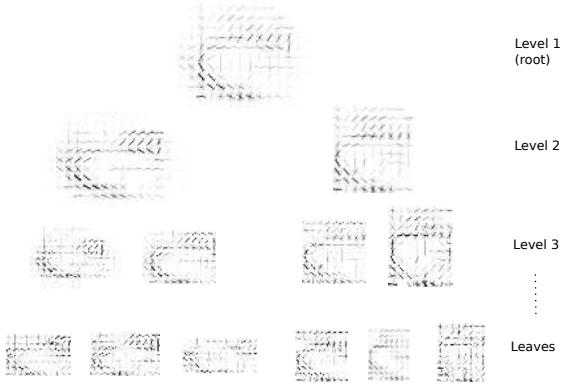


Figure 5. **Hierarchical filter tree.** The discriminative power of the multi-component model is summarized in a single super-node (root). Its left child corresponds to low aspect ratio leaf nodes, whereas the right child captures characteristics of high aspect ratio leaf nodes. The leaves are the individual mixture components.

4.2. Hierarchical pruning with probabilistic bounds

We use the constructed hierarchy to accelerate detection; at any pixel we start at the root, visit the left and right children, check if any of them holds promise for delivering a score above threshold and then accordingly recursively refine or stop. If the score estimate at all nodes upper bounds the leaf scores below it, this procedure is guaranteed to deliver all parts scoring above threshold.

The main technical hurdle is the bound construction. For this we adapt the *probabilistic bounds* [31] used recently in [20] to our coarse-to-fine filter evaluation scheme. While no longer being determinstic, in practice these bounds incur only negligible changes in performance.

In particular, consider having M filters, $f_1, \ldots, f_M$ lying below a node l in the part hierarchy. Given an input HOG feature I, we consider how the average filter $\hat{f} = \frac{1}{M}\sum_{m=1}^{M} f_m$ can be used to bound the individual filter scores, $s_m = \langle f_m, I \rangle$. As we show below, with probability p_e the maximal filter score is bounded from above by the score $\hat{s} = \langle \hat{f}, I \rangle$ of the average filter as follows:

$$\max_{m \in \{1,\ldots,M\}} \langle f_m, I \rangle \leq \langle \hat{f}, I \rangle + \sqrt{\overline{E}(f_1, \ldots, f_M, I)/M p_e}, \quad (3)$$

where $\overline{E}$ combines the local image measurements, I with a measure of distance between the filters f_m and the average filter $\hat{f}$. In particular, we construct an expression for $\overline{E}$ that can be rapidly computed with a single inner product operation; as such the cost of computing the bound for all M filters at test-time does not depend on M.

To prove (3) we proceed as in [20] modelling $\epsilon_m = s_m - \hat{s}$ as a random variable, constructing an interval $[-\alpha, \alpha]$ that contains ϵ_m with high probability, and then bounding s_m from above by $\overline{s} = \hat{s} + \alpha$. The value of α is determined by Chebyshev's inequality:

$$P(|X| > \alpha) \leq V/\alpha^2, \quad (4)$$

which relates the second moment $V = E\{X^2\}$ of a zero-mean random variable X with the probability that its absolute exceeds α. Namely, X lies outside $[-\alpha, \alpha]$ with probability smaller than V/α^2, or, equivalently, X lies in $[-\sqrt{V/p_e}, \sqrt{V/p_e}]$ with probability larger than $1 - p_e$.

Unlike [20], rather than s_m we now need to bound $\max_m s_m$. This requires two modifications: first, we deal with the 'max' operation as follows:

$$P(\max_m s_m > \overline{s}) = P(\vee_m \{s_m > \overline{s}\}) \quad (5)$$

$$\leq \sum_m P(s_m > \overline{s}) < M p_e, \quad (6)$$

where $\vee_m$ indicates a logical-or of the M events, the first inequality follows from the union-bound, and the second inequality holds for a p_e such that $P(s_m > \overline{s}) < p_e, \ \forall l$.

This brings us to our second modification: constructing $\overline{s}$ so that $P(s_m > \overline{s}) < p_e, \forall m$ involves bounding the different variables $s_1, \ldots, s_M$ with a common expression $\overline{s}$. For this we write the scores s_m as summations over HOG cells:

$$s_m = \langle f_m, I \rangle = \sum_c \sum_d f_m(c,d) I(c,d), \quad (7)$$

where $c = (v, h)$ indexes vertical/horizontal positions and d indexes the HOG cell dimensions. We can thus write:

$$\epsilon_m = \sum_c \epsilon_{c,m}, \text{ with } \epsilon_{c,m} = \sum_d \left[\hat{f}(c,d) - f_m(c,d)\right] I(c,d).$$

At any cell c we assume the approximation errors $\hat{f}(c,d) - f_m(c,d)$ can be modelled as independent, identically distributed (iid) variables, and estimate their second moment using the reconstruction error of the respective filter cell: $V_{c,m} = \frac{1}{D} \sum_{d=1}^{D} \left(\hat{f}(c,d) - f_m(c,d)\right)^2$, where $D = 32$ is the HOG cell dimensionality.

Treating $\epsilon_{c,m}$ as the weighted-by-$I(c,d)$ sum of D iid variables its second moment of ϵ_c will be:

$$E\{\epsilon_{c,m}^2\} = V_{c,m} \|I_c\|_2^2, \quad (8)$$

where $\|I_c\|_2^2$ is the ℓ_2 norm of the 32-D vector formed from the c-th HOG cell. We further consider the individual error contributions of the different cells as independent and express the second moment of ϵ_l as follows:

$$E\{\epsilon_m^2\} = \sum_c E\{\epsilon_{c,m}^2\} = \sum_c V_{c,m} \|I_c\|_2^2. \quad (9)$$

The last expression provides us with the error variance needed in Eq. 4 to construct the upper and lower bounds to the score. This however is dependent on the filter index m. In order to drop the dependence on m, we also upper bound the variance in Eq. 9 by maximizing $V_{c,m}$ over m:

$$E\{\epsilon_m^2\} \leq \sum_c \left(\max_m V_{c,m}\right) \|I_c\|_2^2 \doteq \overline{E}. \quad (10)$$

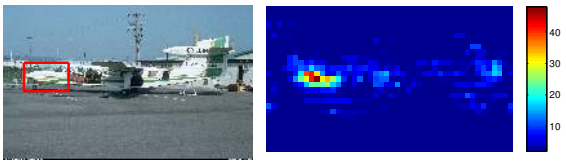


Figure 6. **Pruning of candidate locations.** Detection example and the number of visits by some node in the filter tree. By pruning candidate locations as we move from the root to the leaves, the exact score is evaluated at only a fraction of the image domain(right).

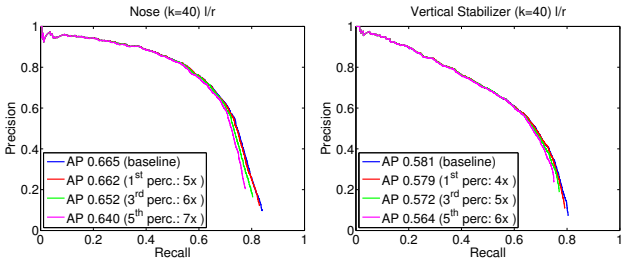


Figure 7. **Cascade detection results.** Precision-recall curves for different percentile values used to determine empirical thresholds, and corresponding speedup with respect to the baseline.

Having derived an expression for the variance that no longer depends on m, we can now bound $s_m, m = 1, \ldots, M$ with a single expression (3), as required.

The computation of Eq. 10 at test time has negligible cost: the cell distortion $\max_m V_{c,m}$ is computed off-line, while the HOG-cell norm $\|I_c\|_2$ is computed once for the HOG pyramid, and reused by all tree nodes. We also note that other than independent errors we did not make further assumptions, which makes our method fairly generic.

4.3. Experiments

We build two tree cascades, one for the left-facing and another for the right-facing filters, each comprising 20 mixture components. Setting the probability of error $p_e = 0.01$ in Eq. 4.2, we obtain upper bounds for the maximal filter score at any node on the filter-tree. Directly using these bounds for bounding-based detection, as outlined above, yields a 3-fold speedup with virtually identical performance as the full-blown convolution (for all results we use identical, single-threaded implementations). We can get an additional speedup by using the bounds of Eq. 4.2 to train empirical pruning thresholds, as in [17]. For this, we set the pruning threshold at every node to reject the bottom-k-th percentile of the training set, for $k = 1, 2, 5$. As shown in Fig. 7, this results in further acceleration (4- to 7- fold), while incurring only a small drop in AP by 0.01-0.02. Our implementation is available at `http://cvn.ecp.fr/personnel/tsogkas/code.html`.

5. Summary

We have introduced AirplanOID, a large dataset of images of planes annotated in detail, discussing its design and

several practical aspects of its construction. This data allowed us to initiate a study of the problem of fine-grained object description, and in particular of the relation between object parts and detailed attribute semantics. We have shown that attributes are often predicted best by the part containing direct evidence about them, but not always due to the existence of contextual ties that are often neglected in attribute modeling. We have also shown that semantic supervision and rich appearance models can improve part detection and hence attribute prediction. Finally, we have introduced a coarse-to-fine technique to detect efficiently these richer part models in images.

Acknowledgments. This research is based on work done at the 2012 CLSP Summer Workshop, and was partially supported by NSF Grant #1005411, ODNI via the JHU HLTCOE and Google Research. S. Tsogkas and I. Kokkinos were supported by ANR-10-JCJC-0205. S. Mahendran was supported by NSF Grant #11-1218709. All the images in our dataset were kindly provided by Mick Bajcar, who very generously agreed to allow the community to use his images *exclusively for non-commercial research purposes*, provided that a suitable *copyright note* is included.

References

[1] Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7), 1999. 6
[2] M. Andreetto, L. Zelnik-Manor, and P. Perona. Unsupervised learning of categorical segments in image collections. 2008. 6
[3] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010. 3
[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *ICCV*, 2009. 5
[5] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011. 4
[6] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, 2013. 3
[7] J. Deng, S. Satheesh, A. Berg, and L. Fei-Fei. Fast and balanced: Efficient label tree learning for large scale object recognition. In *NIPS*, 2011. 6
[8] J. Donahue and K. Grauman. Annotator rationales for visual recognition. In *ICCV*, 2011. 3, 5
[9] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*, 2011. 3, 5
[10] I. Endres, A. Farhadi, D. Hoiem, and D. A. Forsyth. The benefits and challenges of collecting richer object annotations. In *Proc. ACVHL Wokrshop (with CVPR)*, 2010. 3
[11] M. Everingham, A. Zisserman, C. Williams, and L. V. Gool. The PASCAL visual obiect classes challenge 2007 (VOC2007) results. Technical report, Pascal Challenge, 2007. 3
[12] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010. 3
[13] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2, 3, 5
[14] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010. 2, 3, 5
[15] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. *ICCV*, 2003. 3
[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010. 1
[17] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010. 7
[18] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/~rbg/latent-release5/. 4, 5
[19] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*, 2012. 4
[20] I. Kokkinos. Shufflets: shared mid-level parts for fast object detection. In *ICCV*, 2013. 6, 7
[21] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, 2012. 3, 5
[22] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating image descriptions. In *CVPR*, 2011. 2, 3
[23] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *ECCV*, 2008. 3
[24] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009. 3
[25] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 3, 5
[26] C. H. Lampert. Detecting objects in large image collections and videos by *efficient subimage retrieval*. In *ICCV*, 2009. 3
[27] S. Maji. Discovering a lexicon of parts and attributes. In *ECCV Workshop*, 2012. 3
[28] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 2
[29] K. Matzen and N. Snavely. Nyc3dcars: A dataset of 3d vehicles in geographic context. In *ICCV*, 2013. 5
[30] M. Mitchell, J. Dodge, A. Goyal, K. Yamaguchi, K. Stratosi, X. Han, A. Mensch, A. Berg, T. Berg, and H. D. III. Midge: Generating image descriptions from computer vision detections. *EACL*, 2012. 2, 3, 5
[31] M. Mitzenmacher and E. Upfal. *Probability and computing - randomized algorithms and probabilistic analysis*. CUP, 2005. 6
[32] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 3
[33] D. Parikh and G. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*, 2011. 3, 5
[34] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 3
[35] A. Parkash and D. Parikh. Attributes for classifier feedback. In *ECCV*, 2012. 3
[36] O. Russakovsky and L. Fei-Fei. Attribute learning in large-scale datasets. *ECCV Workshop Parts and Attributes*, 2010. 3, 5
[37] B. C. Russel, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. Technical Report AIM-2005-025, MIT AI Lab, 2005. 3
[38] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR*, 2011. 3
[39] L. Torresani, M. Summer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010. 3, 5
[40] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, 2011. 3
[41] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009. 3
[42] G. Wang, D. Forsyth, and D. Hoiem. Comparative object similarity for improved recognition with few or no examples. In *CVPR*, 2010. 3, 5
[43] J. Wang, K. Markert, and M. Everingham. Learning models for object recognition from natural language descriptions. In *BMVC*, 2009. 3
[44] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010. 3, 5
[45] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? BMVA Press, 2012. 2, 5