Open access • Proceedings Article • DOI:10.1109/ICCV.2011.6126398

The truth about cats and dogs — Source link

Omkar M. Parkhi, Andrea Vedaldi, C. V. Jawahar, Andrew Zisserman

Institutions: International Institute of Information Technology, Hyderabad, University of Oxford

Published on: 06 Nov 2011 - International Conference on Computer Vision

Topics: Object detection, Image segmentation and Image texture

The Truth About Cats and Dogs

Omkar M Parkhi[1] Andrea Vedaldi[2] C. V. Jawahar[1] Andrew Zisserman[2]

[1]Center for Visual Information Technology,
International Institute of Information Technology,
Hyderabad 500032, India
{omkar.parkhi@research,jawahar@}iiit.ac.in

[2]Department of Engineering Science,
University of Oxford,
United Kingdom
{vedaldi,az}@robots.ox.ac.uk

Abstract

Template-based object detectors such as the deformable parts model of Felzenszwalb et al. *[11] achieve state-of-the-art performance for a variety of object categories, but are still outperformed by simpler bag-of-words models for highly flexible objects such as cats and dogs. In these cases we propose to use the template-based model to* detect a distinctive part *for the class, followed by detecting the rest of the object via segmentation on image specific information learnt from that part. This approach is motivated by two observations: (i) many object classes contain distinctive parts that can be detected very reliably by template-based detectors, whilst the entire object cannot; (ii) many classes (e.g. animals) have fairly homogeneous coloring and texture that can be used to segment the object once a sample is provided in an image.*

We show quantitatively that our method substantially outperforms whole-body template-based detectors for these highly deformable object categories, and indeed achieves accuracy comparable to the state-of-the-art on the PASCAL VOC competition, which includes other models such as bag-of-words.

1. Introduction

The vast majority of current methods for object category detection use some form of sliding window classifier. In particular, template-based models such as the Deformable Parts Model (DefPM) by [11] currently achieve state-of-the-art performance for the majority of the object classes in international benchmarks such as the PASCAL VOC 2010 [8]. The success of these methods emphasizes the importance of geometry in the description of most visual categories. Yet, for highly flexible and deformable objects such as cats and dogs (figure 1), DefPMs and other template-based models are still outperformed by a large margin by simpler bag-of-words models, which have a much weaker notion of geometry [8]. Not surprisingly, several authors [13, 40] advocates the study of these object categories as prototypical cases for which geometric modeling is challenging.

Figure 1. **The deformable and truncated cat.** Cats exhibit (almost) unconstrained variations in shape and layout. The cat examples shown here are detected by our Distinctive Part Model, but missed by the template based method of [11].

The question we address here is whether it is possible to extend template-based models such as DefPM to be competitive for these highly flexible categories as well. The key insight is that for many objects color and texture are fairly uniform across the entire body, or vary in a manner that can be learnt; and also that many objects have a distinctive part that can be detected well with the current generation of template-based detectors, even though their overall appearance is highly variable. The idea is then to detect first a distinctive part of the category, and second, to segment the category instance primarily using image specific features learnt from that part. We call this a Distinctive Part Model (DisPM, section 2).

For example, for a cat the head is a distinctive part and can be detected well by a template detector such as DefPM. The detected head then provides the cat's fur color and texture, and, in turn, these color/texture distributions can be used to segment out the cat's body. These assumptions are satisfied for instance by numerous animal classes, such as sheep, cows, zebras, horses, elephants. A similar approach can be applied to naked humans (e.g. using face detection to learn an image specific skin color [14]), but clothing renders the model less applicable in this case.

The question is: how well does this DisPM work as a detector? As will be seen (by results on the PASCAL VOC 2010 detection competition [8] in section 3) the per-

formance surpasses existing template models trained on the whole body by far. DisPM is in fact able to detect cats and dogs in quite variable poses, and under considerable partial occlusions and truncations (figure 1).

Related work. Our approach extends template-based detectors such as DefPM, which, by allowing only for limited geometric variability, usually do not work well for highly deformable objects. Similarly, articulated models, such as the pictorial structures [10] typically used for human layout detection, are not appropriate for objects such as cats and dogs as they do not capture the deformation and limb occlusions that they exhibit.

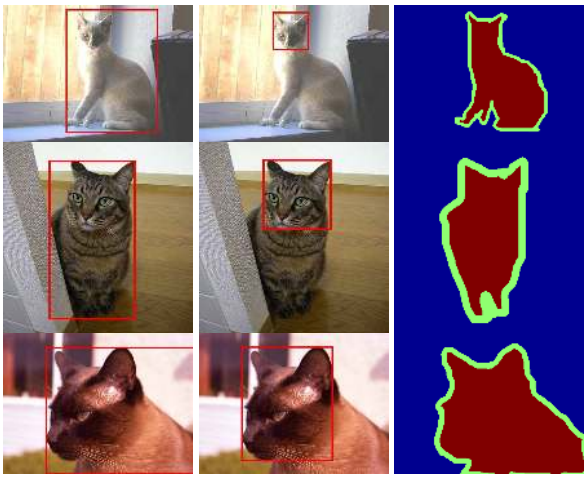
Our method is also directly related to [24] and [40], that have designed and evaluated cat head detectors; section 3 shows DefPM to be a much better detector at this task. Fleuret and Geman [13] have used cats as an example application of their object model based on stationary features. Their coarse-to-fine search strategy uses the cat head as a privileged part, as we do. Unfortunately this algorithm was not evaluated on public benchmarks, making a direct comparison difficult.

Previous work has combined object category detection and segmentation in various ways [16, 19, 22, 33, 34, 36]. However, often the goal of these methods has been segmentation of the entire image, rather than object category detection, whilst others [2, 3, 20, 21, 27, 28, 37] have generally targeted typical views of vehicles and animals (e.g. side views of horses) that are suited to template based detectors. Their aim has not been to handle the variety in appearance and deformation that it is our goal for the new DisPM detector. In fact, a significant difference is that DisPM restricts the use of the template detector to extract just an object part and then leverages on segmentation to extend it to the whole deformable object.

Finally, our work is generally related to sliding-window object detectors. Within the window there may be be a single feature type represented, such as HOG [6] or HOG parts [11], or a bag of visual words [23], or a grid or pyramid of visual words [12, 26], or a combination of such features and kernels [18, 38]. In the recent PASCAL VOC 2010 object detection competition [8] all the top methods were of this kind. There are a number of methods for object detection that start from bottom up segmentation, rather than sliding/jumping windows [1, 15, 17, 25, 29], but they are yet to be competitive with the window based detectors.

2. The distinctive part model

The DisPM extends template-based models to the detection of highly deformable object categories. Consider the case of cats, which we will use as our running (actually sitting) example for describing the new model: extreme articulations, atypical viewpoints, and partial occlusions induce



(g) Object (h) Distinctive Part (i) Trimap

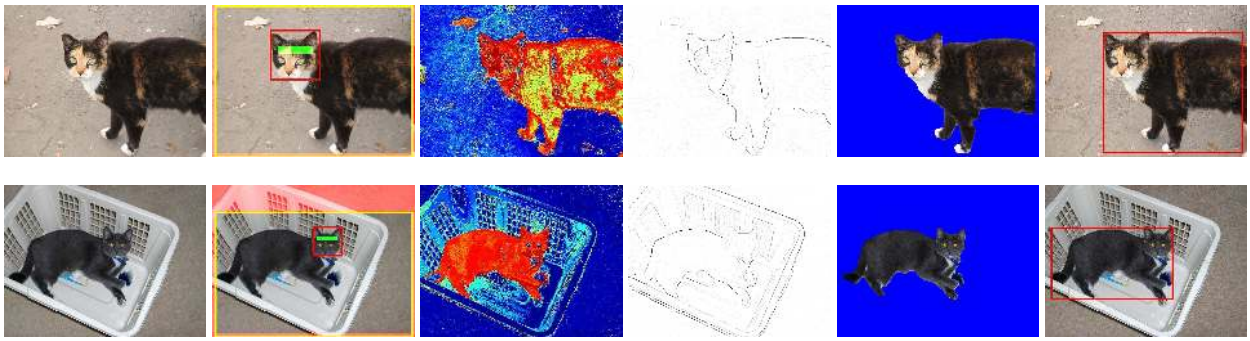
Figure 2. **Annotations.** (a) The PASCAL VOC annotations are tight bounding boxes around the object instances. (b) Additional annotations for the distinctive object part, in this case cat/dog heads. (c) Pixel-level segmentation of the object also provided by PASCAL VOC.

variations of the appearance of a cat that cannot be captured by a template-based model. This is true even for models such as the DefPM detector that account explicitly for deformations of the template.

The DisPM works around this problem by detecting first a stable and distinctive object part, such as the cat head, for which a template-based detector is appropriate. It then uses the detected part to initialize and constrain the segmentation of the rest of the object. DisPM is therefore composed of three elements, illustrated in figure 3: (i) a template-based detector of the distinctive object part, (ii) a model of the object body appearance (color or texture), and (iii) a segmentation algorithm. The segmentation is used here to assist the detection process.

The next three sections describe in detail the three components of the model. For the template-based detector (i) we use the DefPM model based on the implementation publicly available from the author's website (section 2.1). For the local appearance model (ii) we model colors by histograms in RGB space, along with an object boundary detector to aid segmentation (section 2.2). For the segmentation algorithm (iii) we use the standard graph cut model of Boykov *et al*. [5] (section 2.3). Since the appearance model is learned from the object region itself (starting from the distinctive part), graph cut and estimation of the appearance model are alternated to refine the segmentation result (GrabCut [35]).

Training data. We learned and evaluated our model on the PASCAL VOC 2010 detection competition data [8] (note that VOC encourages evaluating detectors specialized on particular object categories as well as general purpose



(a) Original Image (b) Head detection (c) Fg. posteriors (d) Berkeley edges (e) Fg. segmentation (f) Output bnd. box

Figure 3. **Overview of the model.** A distinctive part, the head in this case, is detected using the DefPM model [11]. (b) The detected part ROI (red rectangle) is used to define a search region for the object (yellow rectangle), and also seeds the foreground color distribution (green rectangular region). The background color distribution is learnt from the red area. (c) the foreground posterior, computed using the seed and background data (red is high, blue is low probability). These posteriors form the unary term of the energy function used in segmentation. The pairwise terms use the Berkeley edge detector response (d). A graph cuts binary optimization gives the foreground segmentation (e). The detection result is a tight bounding box around the foreground segment (f).

detectors). The VOC data is a large collection of images with annotations for twenty object classes, including cats and dogs. In particular, the VOC 2010 data contains about 10,000 training and validation images and 10,000 test images. The VOC publishes bounding box annotations and trimaps (figure 2) for the training and validation subsets, while the evaluation on the test data is carried independently by the VOC itself.

Recently Bourdev and Malik [4] advocated the use of manual annotations for training distinctive object parts (poselets). Here we use a similar approach. Specifically, the VOC training and validation data are annotated with bounding boxes by following the same procedure used for the construction of the VOC annotations [9]: for example, a head bounding box is defined as a tight fitting box, containing the face and ears (e.g. in figure 2). These annotations are then used to learn the distinctive part model.

2.1. Part model

The distinctive object part is detected by means of the DefPM. As will be shown in section 3, this model is excellent for structures that are relatively stable, such as, for example, the face of a cat, but is relatively poor for highly deformable objects, such as the cat body. The detected part is used to determine an *image-specific* color model for the cat, and also to predict a (maximal) bounding box for the entire cat.

The DefPM detector is a mixture of templates, each of which is a collection of parts connected by springs. Parts are described by linear filters on top of low level features such as HOG [6] and the model is learned by means of a latent SVM. See section 3 for further details and figure 3(b) for example detections.

2.2. Whole object model

The object appearance model captures the material of the object (color) and the object discontinuities (edges). For the object color, there are two source of information that can be used. First, some colors cannot belong to any of the object instances (e.g. there are no green, blue, or purple cats), which is used to construct a *color prior for the category*. This is learned from the trimap object segmentations (Fig. 2) by computing color histograms of the foreground (cat) and background (non-cat) regions. Second, the color of the specific object *instance* being detected, and of the background scene in which it is found, can be estimated from the distinctive part. For cats and dogs, the head provides a cue on the color of the fur, and image pixels far enough from the head are used to estimate the color of the background.

Category color prior. Colors are modeled by means of histograms. We use a relatively high dimensional histogram $\mathbf{h} \in \mathbb{R}^{32\times32\times32}$ but smooth it by a small Gaussian kernel (of isotropic standard deviation $\sigma = 0.025$) in order to reduce the variance of the estimator. The global foreground/background color histograms $\mathbf{h}^0_{\mathrm{fg}}, \mathbf{h}^0_{\mathrm{bg}}$ are obtained from all the foreground/background regions in the training set.

Instance-specific color. The distinctive part of the object is used to obtain an instance-specific foreground $\mathbf{h}_{\mathrm{fg}}$ and background $\mathbf{h}_{\mathrm{bg}}$ color models. The foreground color is estimated by sampling the pixels contained in the *foreground seed*. The seed is a rectangular sub-region of the distinctive part that is contained in the foreground region with very high probability in the training data. For instance, the foreground seed of cats roughly corresponds to the fore-

head. The location of this region inside the distinctive part is learnt from the training data. The background color is estimated from the pixels that are outside a *maximal bounding box*, i.e. a bounding box that contains almost surely the entire object. The maximal bounding box is obtained by aligning and scaling a template box to the rectangle of the distinctive part detection. The dimensions of the template itself are learned by requiring it to be the smallest box that contains 99% of the object pixels for all training images. To handle the case where no part of the image is inside the maximal bounding box, a thin strip of pixels around the image (20 pixels wide) is always included to estimate the background color. Examples of the seed and of the bounding box are shown in figure 3(b) (these regions will be used in section 2.3 to further constrain the segmentation geometrically).

Foreground and background posteriors. Let $\mathbf{x}$ be an image and $\mathbf{y}$ be a partition of the image into foreground (object) and background components. In particular, let $\mathbf{x}_i \in \mathbb{R}^3$ denote the color of the i-th pixel (in RGB space) and let y_i be equal to $+1$ if the pixel belongs to the object and to -1 otherwise. Given the color histogram $\mathbf{h}_{\mathrm{fg}}, \mathbf{h}_{\mathrm{bg}}, \mathbf{h}^0_{\mathrm{fg}}, \mathbf{h}^0_{\mathrm{bg}}$, we can define three likelihoods:

$$p(\mathbf{x}|y=+1,\mathrm{fg}) = \mathbf{h}_{\mathrm{fg}}(\mathbf{x}), \quad p(\mathbf{x}|y=-1,\mathrm{bg}) = \mathbf{h}_{\mathrm{bg}}(\mathbf{x}),$$
$$p(\mathbf{x}|y=+1,\mathrm{fg}^0) = \mathbf{h}^0_{\mathrm{fg}}(\mathbf{x}), \quad p(\mathbf{x}|y=-1,\mathrm{bg}^0) = \mathbf{h}^0_{\mathrm{bg}}(\mathbf{x}),$$

fg and bg are foreground background pixels from the given image, and fg^0 and bg^0 are foreground and background pixels from the set of training images. By assuming $P[y=+1] = P[y=-1] = 1/2$, these are combined into two posteriors

$$p_1(y|\mathbf{x}) = \frac{p(\mathbf{x}|y=+1,\mathrm{fg})}{p(\mathbf{x}|y=+1,\mathrm{fg}) + p(\mathbf{x}|y=-1,\mathrm{bg})}, \quad (1)$$

$$p_2(y|\mathbf{x}) = \frac{p(\mathbf{x}|y=+1,\mathrm{fg}^0)}{p(\mathbf{x}|y=+1,\mathrm{fg}^0) + p(\mathbf{x}|y=-1,\mathrm{bg}^0)}. \quad (2)$$

The first one discriminates between the color of the object instance and the color of its surrounding (as estimated from the seed and the maximal bounding box), and the second one between that of the object and of generic clutter. The latter helps eliminating impossible colors (e.g. green cats) that may not be sampled outside the maximal object bounding box. These two are combined into a unique posterior by additive combination ($p(y|\mathbf{x}) \propto c_1 p_1(y|\mathbf{x}) + c_2 p_2(y|\mathbf{x})$ where the weights c_i are learnt from validation data (and have the values $c_1 = 1/10, c_2 = 9/10$). Example foreground posteriors, $p(y=1|\mathbf{x})$, are shown in figure 3(c).

Modeling edges. In addition to color, the model also uses an edge detector in order to further improve the quality of the final object segmentation. The edge map will be used to encourage the segmentation boundaries to match discontinuities of image edges. In this work we leverage on the powerful Berkeley PB edge detector [30]. Compared to other detectors such as Canny, PB is designed to suppress intensity discontinuities which correspond to texture rather than actual object boundaries. See figure 3(d).

2.3. Segmentation model

Once the distinctive object part has been detected, it must be extended to a segmentation of the entire object (see figure 3(e)). As we expect the object to be highly deformable but to have a distinctive material, this can be achieved by a well designed segmentation algorithm.

For segmentation we use a graph cut [5] based energy minimization formulation. The cost function is given by

$$E(\mathbf{x},\mathbf{y}) = -\sum_i \log p(y_i|x_i) + \sum_{(i,j)\in\mathcal{E}} S(y_i,y_j|\mathbf{x}) \quad (3)$$

The edge system $\mathcal{E}$ determines the pixel neighborhoods and here is the standard eight-way connectivity scheme. The pairwise potential $S(y_i,y_j|\mathbf{x})$ favors neighbor pixels to have the same label unless a PB edge separates them:

$$S(y_i,y_j|\mathbf{x}) = \gamma \exp(-e_j(\mathbf{x})/\beta) \quad (4)$$

where $e_j(\mathbf{x})$ is the PB edge intensity at pixel j and $\beta = \langle e_j(\mathbf{x}) \rangle$ is the average edge intensity in the image. Note that the edge is measured only at pixel j, as defined by the edge system $\mathcal{E}$ (here j is the pixel more on the right/south). The parameter γ is learnt on the validation data.

The distinctive part detection is used to fix the values of some labels $\mathbf{y}$ (clamping) as follows: (i) the foreground seed region must be labeled as foreground, and (ii) the region outside the maximal bounding box must be background. These two regions were defined above in section 2.2.

The segmentation is defined as the minimizer $\arg\min_{\mathbf{y}} E(\mathbf{x},\mathbf{y})$ of the energy using graph cut. In fact, since the color of foreground and background can be estimated more accurately as a better segmentation of the object becomes available, GraphCut is alternated to re-estimate the color model, in the manner of GrabCut [35]. In section 3 we show that initializing from the posteriors of section 2.2, yields a substantial improvement in detection performance over simply initializing from the clamped regions.

Cleaning-up and detection. Given the segmentation result from GrabCut, this is cleaned-up by preserving only the connected foreground component that intersects with the distinctive part and discarding the others. The final object bounding box is estimated as the smallest box that fully contains the segmented foreground region (see figure 3(f)).

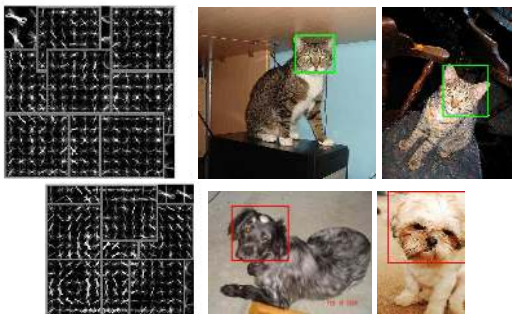


Figure 4. **Distinctive part detector**. First row: The DefPM model [11] for the cat head, used as distinctive part, and example detections. Second row: the same for dog.

The detector score is obtained from a combination of the DefPM score and size of the distinctive part detection.

3. Results

Following the PASCAL VOC best practices [8], the various components of the model are evaluated in detail on the PASCAL VOC 2010 train/validation sets and overall results for the complete model are given on the test set to allow for a direct comparison with other published methods. Since we use head annotations to train the distinctive part model, we evalute our results against the VOC detection competition 4, which allows additional annotations.

The performance of a detector is evaluated in term of the Average Precision (AP) of the ranked list of detections, where a detection is considered to be correct if its overlap ratio with a ground truth bounding box is at least 0.5 and if it is not a duplicate (see [9] for details).

Learning the distinctive part. The distinctive part annotations are used to train a DefPM model for the part (figure 4) with one aspect, eight high resolution parts, and a low resolution one (root filter). The low level image features are HOG [6, 11] (capturing shape) and LBP [31] (capturing texture). The DefPM detector supports multiple components, but in our experiments we use a single one as we found empirically that this worked better in our case. Figure 4 shows examples of the detected cat/dog heads with variations in pose, appearance, and size.

Precision-recall curves for the DefPM detector for the cat heads in the VOC 2010 validation data are given in figure 5(b). With the standard PASCAL VOC overlap ratio of 0.5, the detector AP is 45% with HOG features only, and this improves to 49% when the LBP features are added. Since the DisPM uses the distinctive part as a seed to obtain a segmentation for the whole object, a less strict (than 0.5) overlap ratio often suffices for this purpose (as will be seen below). Thus it is interesting to note that for a (looser) overlap ratio of 0.2, the AP of the head detector is 61% with a recall of 80%. The recall-precision curve for this overlap ratio is also shown in figure 5(b). The DefPM performs much better than alternative cat head detectors available in the literature. Specifically, when trained and tested on the VOC 2007 cat heads, DefPM achieves an AP of 54.6%, while the detector of Zhang *et al*. [40] obtains 34.4% with the same data. The detector of Laptev [24] obtains 18.7% on the VOC 2007 test data (when trained on the VOC 2006 training data).

Whole object detectors: baselines. The first baseline is the standard DefPM model trained to detect the whole object. For cats, training on the VOC training data and testing on the validation data gives an AP of just 29% (figure 5(a)). Based on the PASCAL VOC 2010 results, the performance of the newest DefPM version (which is not yet available to the public) is, on the VOC 2010 test data, about the same (31.8%). This level of performance is relatively poor compared to other classes (e.g. the performance of the DefPM detector on the VOC vehicles is around 50% AP). The model does not seem capable of capturing the variability of the cat *bodies*. To verify this, consider as a second baseline a simple head-to-cat regressor. This regressor is obtained by computing the average ratios between the size of the cat head and the margins between the cat head bounding box and the bounding box of the whole cat. These ratios are then used to predict a bounding box for the cat given a novel head detection. This simple head-to-cat regressor has 31.1% AP, which *already exceeds the performance of the DefPM detector trained on the whole cat*.

Whole object detectors: upper-bounds. Given the detections for the distinctive part, it is easy to compute an upper bound for the performance of the DisPM by mapping each part detection to its corresponding ground-truth object bounding box, if there is one (we say that the part corresponds to the object if more than 50% of the area of its bounding box is included in the object bounding box). In this way, one obtains a cat detector with AP of 67% (figure 5(c)). While this is an ideal result, it is worth noting that the performance is more than twice that of the standard DefPM detector.

Postprocessing. All the top methods in the PASCAL Challenge [8] rerank detections based on global image cues and other statistics. In our case, the final scoring for a candidate detection is obtained by combining, by means of a linear SVM, the following seven features. The first feature is the DefPM score for the distinctive part; the second and third features are the output of image-level bag-of-word classifiers [39], trained to detect cats and dogs respectively (the inclusion of both animals helps disambiguating between them, similarly to [11]); the fourth and fifth features are also two global cat and dog scores, obtained as

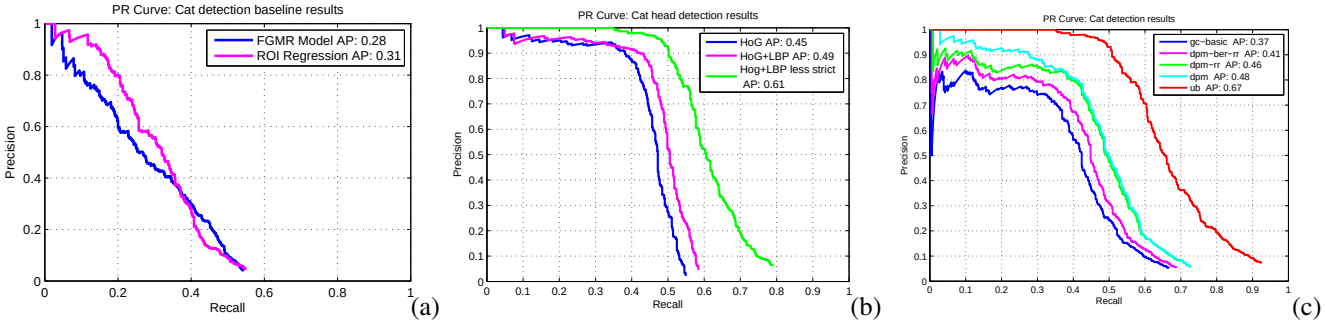


Figure 5. **Performance of the model components for cat detection on the VOC2010 Validation data.** (a) Baseline cat ROI detection results – the DefPM model (trained on the whole object) and regression on the head detections. (b) Head ROI detection results using the DefPM model (trained only on heads) with and without LBP. (c) Components of the DisPM model: `gc-basic` (GrabCut initialized from the clamped regions, without Berkeley edges nor reranking), `dpm-ber-rr` (as previous case, but GrabCut with the posterior from Sect. 2.2), `dpm-rr` (with Berkeley edges), `dpm` (with reranking). Finally, *ub* shows the upper bound on the detection AP.

the maximum response of the DefPM detectors within each image. The last two features are the size of the detection relative to the image size and its aspect ratio, which capture weak pose information. Postprocessing improves the results by 2-3% AP points, a similar gain was noted by [11].

Results on cats and dogs. Having defined and measured upper and lower bounds (from the baselines), we now turn to the performance of the DisPM itself. This is shown in figure 5(c), where the contribution of the various components of the model are detailed *for the VOC 2010 validation data*: (i) the most basic (damaged) form of the model is to segment using GrabCut but with the foreground and background regions defined only by the clamped areas, and without using the Berkeley edge detector (instead the pair wise term (4) measures neighboring image intensity differences directly as in [5]). This is shown as `gc-basic` and has an AP of only 37%. Adding in the posterior computation from section 2.2 to initialize the GrabCut (`dpm-ber-rr`) increases the AP to 41%. A further increase is obtained by using Berkeley edges instead of image differences in (4), and the performance reaches 46% (`dpm-rr`). Finally, the full DisPM including the reranking step (`dpm`) achieves 48%, which surpasses the baselines (DefPM and regressor) by about 20% AP. A similar analysis holds for dogs, for which the final AP of the DisPM detector is 36%, which also about 20% better than both the baselines (the upper bound being 51%). While the performance of the DisPM exceeds both the baselines by a wide margin, there is still a significant gap to the upper bound. We describe the reasons for this gap below, and in section 4 discuss how the gap can be reduced. Examples detections are shown in figures 7 and 8 for cats and dogs respectively.

Finally, on the VOC 2010 *test* data the performance of the cat and dog detectors are respectively 45.3% and 36.8%, both of which improve significantly on the latest DefPM results (31.8% and 21.5%) and are very close to the state of the art (47.7% and 37.2%) [8].

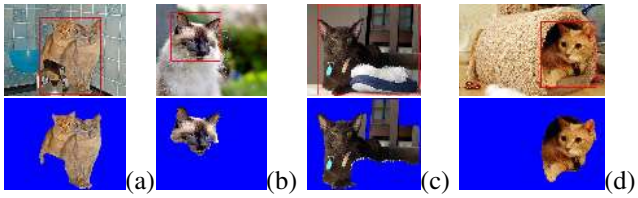


Figure 6. **Failure cases of the DisPM detector**. Top row: predicted detection bounding box superimposed on the image. Bottom row: the foreground segmentation. Failure modes (details in section 3): (a) multiple cats, (b) head and body of different color, (c) background and cat of the same color, (d) disconnected cat region (see paw on the left).

Failure modes. Figure 6 gives examples of the algorithm failing. The principal reason for failure is that the foreground head seed is not able to predict the body color adequately. This is because the body has varying brightness due to shadows or highlights, or because the fur has two different colors but the head and body have different proportions of these. Other less common failures are due to multiple cats, background bleeding, or foreground occlusion where the cat is divided into several unconnected components.

4. Conclusion and discussion

Given the current performance of the DisPM detector, the truth about cats and dogs is that starting from a distinctive part it is possible to detect far more and varied instances than can be obtained with a whole body template detector. Indeed, the DisPM detector is comparable to the state of the art. This is remarkable – a simple model using only two feature types (HOG and LBP for the distinctive part) and image specific color, matches the performance of algorithms using multiple features, including pyramids and kernels (e.g. the PASCAL VOC 2010 winner for this class).

However, to improve the DisPM performance further

Figure 7. **Cat detections.** A sample of the detections and segmentations produced by the DisPM detector (VOC 2010 validation data). It can be seen that cats are successfully detected despite having different fur colors, and appearing in a variety of postures etc.

will require using more hints, cues and constraints in the segmentation model. For example: (i) Class based edge classification – learning which of the edges are due to the cat silhouette edges, and which arise from other sources

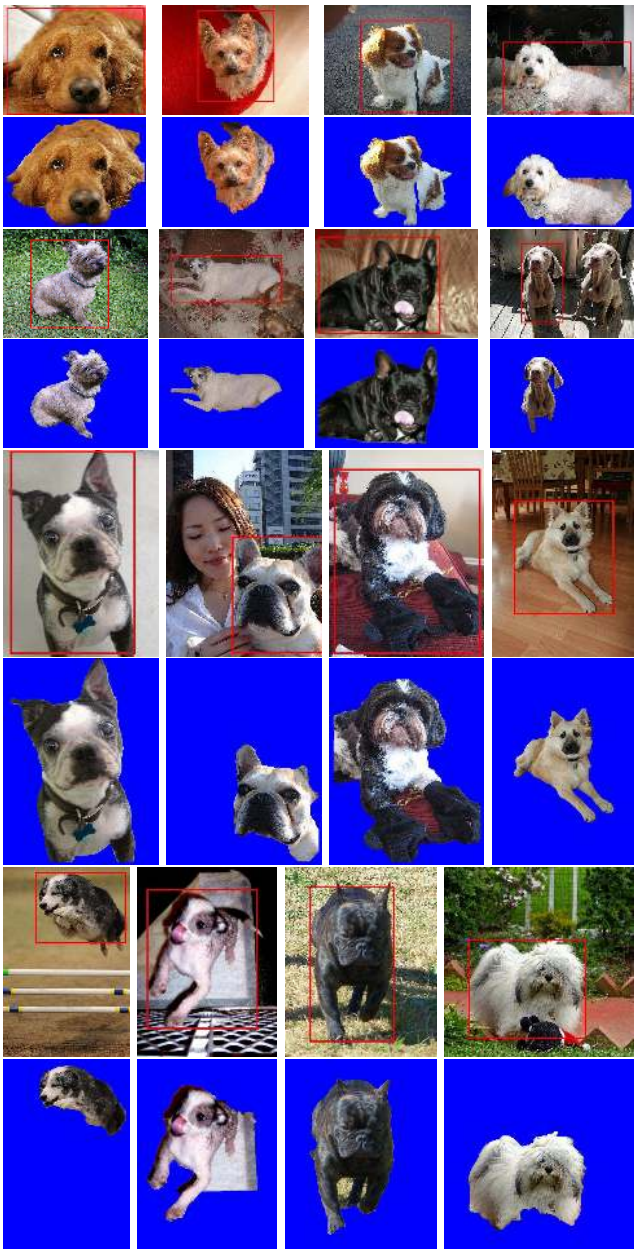


Figure 8. **Dog detections.** A sample of the detections produced by the DisPM detector (VOC 2010 validation data).

(e.g. an occlusion boundary of a chair). Others have learnt edges for classes quite successfully [7, 32]. (ii) Class specific color restrictions – For example that cat coloring is uni or bi modal, e.g. only grey or black and white. (iii) Class specific shape restrictions – that parts of the boundary should be smooth and curved.

Although we have primarily investigated the DisPM detector for a subset of the animals of the PASCAL VOC challenge, there is no doubt that the distinctive part approach is applicable to many other animal classes.

Acknowledgments. We are grateful for financial support

from the UKIERI, ONR MURI N00014-07-1-0182 and ERC grant VisRec no. 228180.

References

[1] N. Ahuja and S. Todorovic. Connected segmentation tree – a joint representation of region layout and hierarchy. In *Proc. CVPR*, 2008.

[2] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proc. ECCV*, pages 109–124, 2002.

[3] E. Borenstein and S. Ullman. Learning to segment. In *Proc. ECCV*, volume 3, pages 315–328, 2004.

[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *Proc. ICCV*, 2009.

[5] Y. Boykov, H. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), 2001.

[6] N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In *Proc. CVPR*, volume 2, pages 886–893, 2005.

[7] P. Dollár, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *Proc. CVPR*, 2006.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results, 2010.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 88(2):303–338, Jun 2010.

[10] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005.

[11] P. F. Felzenszwalb, R. B. Grishick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 2009.

[12] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *Proc. ICCV*, 2005.

[13] F. Fleuret and D. Geman. Stationary features and cat detection. *Journal of Machine Learning Research*, 9, 2008.

[14] J. Fritsch, S. Lang, M. Kleinehagenbrock, G. A. Fink, and G. Sagerer. Improving adaptive skin color segmentation by incorporating results from face detection. In *IEEE Workshop on Robot and Human Interactive Communication*, 2002.

[15] J. M. Gonfaus, X. Boix, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzàlez. Harmony potentials for joint classification and segmentation. In *Proc. CVPR*, 2011.

[16] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *Proc. ICCV*, 2009.

[17] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik. Recognition using regions. In *Proc. CVPR*, 2009.

[18] H. Harzallah, C. Schmid, F. Jurie, and A. Gaidon. Classification aided two stage localization. Presented at the ECCV VOC 2008 Workshop, 2008.

[19] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *Proc. ECCV*, 2008.

[20] D. Hoeim, C. Rother, and J. M. Winn. 3D layout CRF for multi-view object class recognition and segmentation. In *Proc. CVPR*, 2008.

[21] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *Proc. CVPR*, volume 1, pages 18–25, 2005.

[22] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. Torr. Where, what and how many? combining object detectors and CRFs. In *Proc. ECCV*, 2010.

[23] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE PAMI*, 2009.

[24] I. Laptev. Improvements of object detection using boosted histograms. In *Proc. BMVC*, 2006.

[25] D. Larlus and F. Jurie. Combining appearance models and Markov random fields for category level object segmentation. In *Proc. CVPR*, 2008.

[26] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proc. CVPR*, 2006.

[27] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, May 2004.

[28] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *Proc. ECCV*, 2006.

[29] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *Proc. CVPR*, 2010.

[30] D. R. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 1, 2004.

[31] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29, 1996.

[32] M. Prasad, A. Zisserman, A. W. Fitzgibbon, M. P. Kumar, and P. H. S. Torr. Learning class-specific edges for object detection and segmentation. In *Proc. ICVGIP*, Dec 2006.

[33] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Proc. ICCV*, 2007.

[34] D. Ramanan. Using segmentation to verify object hypotheses. In *Proc. CVPR*, 2007.

[35] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *Proc. ACM SIGGRAPH*, 23(3):309–314, 2004.

[36] B. C. Russel, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proc. CVPR*, 2006.

[37] J. Shotton, J. Winn, C. Rother, and A. Criminisi. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proc. ECCV*, 2006.

[38] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proc. ICCV*, 2009.

[39] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *Proc. NIPS*, 2009.

[40] W. Zhang, J. Sun, and X. Tang. Cat head detection - how to effectively exploit shape and texture features. In *Proc. ECCV*, 2008.