Deformable Part-based Fully Convolutional Network for Object Detection

Taylor Mordan[1, 2]
taylor.mordan@lip6.fr
Nicolas Thome[3]
nicolas.thome@cnam.fr
Matthieu Cord[1]
matthieu.cord@lip6.fr
Gilles Henaff[2]
gilles.henaff@fr.thalesgroup.com

[1] Sorbonne Universités
UPMC Univ. Paris 06, CNRS, LIP6 UMR 7606
4 Place Jussieu, 75005 Paris, France

[2] Thales Optronique S.A.S.
2 Avenue Gay-Lussac, 78990 Élancourt, France

[3] CEDRIC
Conservatoire National des Arts et Métiers
292 Rue St Martin, 75003 Paris, France

Abstract

Existing region-based object detectors are limited to regions with fixed box geometry to represent objects, even if those are highly non-rectangular. In this paper we introduce DP-FCN, a deep model for object detection which explicitly adapts to shapes of objects with deformable parts. Without additional annotations, it learns to focus on discriminative elements and to align them, and simultaneously brings more invariance for classification and geometric information to refine localization. DP-FCN is composed of three main modules: a Fully Convolutional Network to efficiently maintain spatial resolution, a deformable part-based RoI pooling layer to optimize positions of parts and build invariance, and a deformation-aware localization module explicitly exploiting displacements of parts to improve accuracy of bounding box regression. We experimentally validate our model and show significant gains. DP-FCN achieves state-of-the-art performances of 83.1% and 80.9% on PASCAL VOC 2007 and 2012 with VOC data only.

1 Introduction

Recent years have witnessed a great success of Deep Learning with deep Convolutional Networks (ConvNets) [19, 20] in several visual tasks. Originally mainly used for image classification [17, 19, 35], they are now widely used for others tasks such as object detection [5, 13, 14, 24, 41] or semantic segmentation [3, 21, 26]. In particular for detection, region-based deep ConvNets [5, 13, 14] are currently the leading methods. They exploit region proposals [11, 28, 29] as a first step to focus on interesting areas within images, and then classify and finely relocalize these regions at the same time.

Although they yield excellent results, region-based deep ConvNets still present a few issues that need to be solved. Networks are usually initialized with models pre-trained on ImageNet dataset [30] and are therefore prone to suffer from mismatches between classification and detection tasks. As an example, pooling layers bring invariance to local transformations and help learning more robust features for classification, but they also reduce the

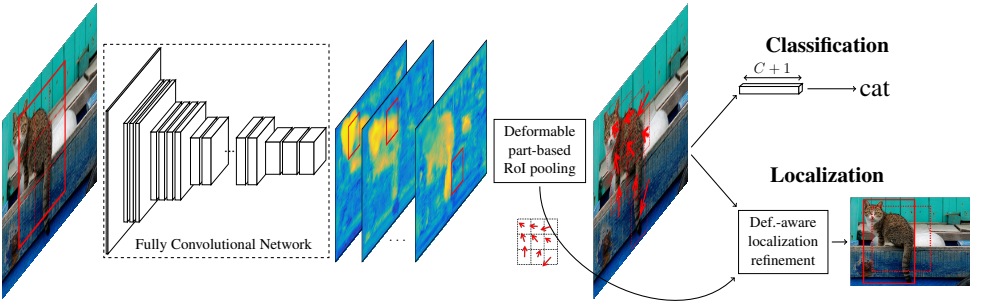


Figure 1: **Architecture of DP-FCN.** It is composed of a FCN to extract dense feature maps with high spatial resolution (Section 3.1), a deformable part-based RoI pooling layer to compute a representation aligning parts (Section 3.2) and two sibling classification and localization prediction branches (Section 3.3). Initial rectangular region is deformed to focus on discriminative elements of object. Alignment of parts brings invariance for classification and geometric information refining localization *via* a deformation-aware localization module.

spatial resolution of feature maps and make the network less sensitive to the positions of objects within regions [5], both of which are bad for accurate localization. Furthermore, the use of rectangular bounding boxes limits the representation of objects, in the way that boxes may contain a significant fraction of background, especially for non-rectangular objects.

Before the introduction of Deep Learning into object detection with [14], state of the art was led by approaches exploiting Deformable Part-based Models (DPMs) [8]. These methods are in contrast with region-based deep ConvNets: while the latter relies on strong features learned directly from pixels and exploit region proposals to focus on interesting areas of images, DPM explicitly takes into account geometry of objects by optimizing a graph-based representation and is usually applied in a sliding window fashion over images. Both approaches exploit different hypotheses and seem therefore complementary.

In this paper, we propose Deformable Part-based Fully Convolutional Network (DP-FCN), an end-to-end model integrating ideas from DPM into region-based deep ConvNets for object detection, as an answer to the aforementioned issues. It learns a part-based representation of objects and aligns these parts to enhance both classification and localization. Training is done with box-level supervision only, *i.e*. without part annotation. It improves upon existing object detectors with two key contributions.

The first one is the introduction of a new deformable part-based RoI pooling layer, which explicitly selects discriminative elements of objects around region proposals by simultaneously optimizing latent displacements of all parts (middle of Fig. 1). Using a fixed box geometry must be sub-optimal, especially when objects are not rigid and parts can move relative to each others. Through alignment of parts, deformable part-based RoI pooling increases the limited invariance to local transformations brought by pooling, which is beneficial for classification.

Aligning parts also gives access to their configuration (*i.e*. their positions relative to each others), which brings important geometric information about objects, *e.g*. their shapes, poses or points of view. The second improvement is the design of a deformation-aware localization module (right of Fig. 1), a specific module exploiting configuration information to refine

localization. It improves bounding boxes regression by explicitly modeling displacements of parts within the localization branch, in order to tightly fit boxes around objects.

By integrating previous ideas into Fully Convolutional Networks (FCNs) [5, 17] (left of Fig. 1), we obtain state-of-the-art results on standard datasets PASCAL VOC 2007 and 2012 [7] with VOC data only. We show that those architectures are amenable to an efficient computation of parts and their deformations, and offer natural solutions to keep spatial resolution. The application of deformable part-based approaches is in particular severely dependent on the availability of rather fine feature maps [15, 31, 36].

2 Related work

Region-based object detectors. Region-based deep ConvNets are currently the leading approach in object detection. Since the seminal works of R-CNN [14] and Fast R-CNN [13], most of object detectors exploit region proposals or directly learn to generate them [11, 28, 29]. Compared to sliding window approach, the use of region proposals allows the model to focus computation on interesting areas of images and to balance positive and negative examples to ease learning. Other improvements are now commonly used, *e.g.* using intermediate layers to refine feature maps [2, 18, 24, 41] or selecting interesting regions for building mini-batches [5, 32].

Deformable Part-based Models. The core idea of DPM [8] is to represent each class by a root filter describing whole appearances of objects and a set of part filters to finely model local parts. Each part filter is assigned to an anchor point, defined relative from the root, and move around during detection to model deformations of objects and best fit them. A regularization is further introduced in the form of a deformation cost penalizing large displacements. Each part is then optimizing a trade-off between maximizing detection score and minimizing deformation cost. Final output combines scores from root and all parts. Accurate localization is done with a post-processing step.

Several extensions have been proposed to DPM, *e.g.* using a second hierarchical level of parts to finely describe objects [44], sharing part models between classes [27], learning from strongly supervised annotations (*i.e.* at the part level) to get a better model [1], exploiting segmentation clues to improve detection [9].

Part-based deep ConvNets. The first attempts to use deformable parts with deep ConvNets simply exploited deep features learned by an AlexNet [19] to use them with DPMs [15, 31, 36], but without region proposals. However tasks implying spatial predictions (*e.g.* detection, segmentation) require fine feature maps in order to have accurate localization [24]. The fully connected layers were therefore discarded to keep enough spatial resolution, lowering results. We solve this issue by using a FCN, well suited for these kinds of applications as it naturally keeps spatial resolution. Thanks to several tricks easily integrable into FCNs (*e.g.* dilated convolutions [3, 26, 39] or skip pooling [2, 18, 41]), FCNs have recently been successful in various tasks, *e.g.* image classification [17, 38, 40], object detection [5], semantic segmentation [21], weakly supervised learning [6].

[43] introduces parts for detection by learning part models and combining them with geometric constraints for scoring. It is learned in a strongly supervised way, *i.e.* with part annotations. Although manually defining parts can be more interpretable, it is likely suboptimal for detection as they might not correspond to most discriminative elements.

Parts are often used for fine-grained recognition. [22] proposes a module for localizing and aligning parts with respect to templates before classifying them, [34] finds part proposals from activation maps and learns a graphical model to recognize objects, [42] uses two sub-networks for detection and classification of parts, [33] considers parts as a vocabulary of latent discriminative features decoupled from the task and learns them in an unsupervised way. Usage of parts is also common in semantic segmentation, *e.g.* [4, 21, 37].

The work closest to ours is R-FCN [5], which also uses a FCN to achieve a great efficiency. We improve upon it by learning more flexible representations than with fixed box geometry. It allows our model to align parts of objects to bring invariance into classification and to exploit geometric information from positions of parts to refine localization.

3 Deformable Part-based Fully Convolutional Networks

In this section, we present Deformable Part-based Fully Convolutional Network (DP-FCN), a deep network for object detection. It represents regions with several parts that it aligns by explicitly optimizing their positions. This alignment improves both classification and localization: the part-based representations are more invariant to local transformations and the configurations of parts give important information about the geometry of objects. This idea can be inserted into most of state-of-the-art network architectures. The model is end-to-end trainable without part annotation and adds a small computational overhead only.

The complete architecture is depicted in Fig. 1 and is composed of three main modules: (i) a Fully Convolutional Network (FCN) applied on whole images, (ii) a deformable part-based RoI pooling layer, and (iii) two sibling prediction layers for classification and localization. We now describe all three parts of our model in more details.

3.1 Fully convolutional feature extractor

Our model relies on a FCN (*e.g.* [17, 38, 40]) as backbone architecture, as this kind of network enjoys several practical advantages, leading to several successful models, *e.g.* [5, 6, 21]. First, it allows to share most computation on whole images and to reduce per-RoI layers, as noted in R-FCN [5]. Second and most important to our work, it directly provides feature maps linked to the task at hand (*e.g.* detection heatmaps, as illustrated in the middle of Fig. 1 and on the left of Fig. 2) from which final predictions are simply pooled, as done in [5, 6]. Within DP-FCN, inferring the positions of parts for a region is done with a particular kind of RoI pooling that we describe in Section 3.2.

The fully convolutional structure is therefore suitable for computing responses of all parts for all classes as a single map for each of them. A corresponding structure is used for localization. The complete representation for a whole image (classification and localization maps for each part of each class) is obtained with a single forward pass and is shared between all regions of the same image, which is very efficient.

Since relocalization of parts is done within feature maps, the resolution of those maps is of practical importance. FCNs contain only spatial layers and are therefore well suited for preserving spatial resolution, as opposed to networks ending with fully connected layers, *e.g.* [19, 35]. Specifically, if the stride is too large, deformations of parts might be too coarse to describe objects correctly. We reduce it by using dilated convolutions [3, 26, 39] on the last convolution block and skip pooling [2, 18, 41] to combine the last three.

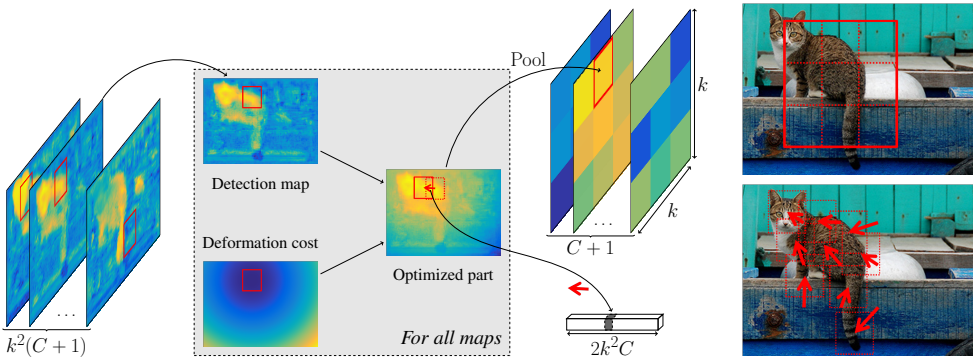


Figure 2: **Deformable part-based RoI pooling (left).** Each input feature map corresponds to a part of a class (or background). Positions of parts are optimized separately within detection maps with deformation costs as regularization, and values are pooled within parts at the new locations. Output includes a map for each class and the computed displacements of parts, to be used for localization. **Illustration of deformations (right).** Parts are moved from their initial positions to adapt to the shape of the object and better describe it.

3.2 Deformable part-based RoI pooling

The aim of this layer is to divide region proposals R into several parts and to locally relocalize these to best match shapes of objects (see Fig. 2). Each part then models a discriminative local element and is to be aligned at the corresponding location within the image. This deformable part-based representation is more invariant to transformations of objects because the parts are positioned accordingly and their local appearances are stable [8]. This is especially useful for non-rigid objects, where a box-based representation must be sub-optimal.

The separation into parts is done with a regular grid of size $k \times k$ fitted to regions [5, 13]. Each cell (i, j) is then interpreted as a distinct part $R_{i,j}$. This strategy is simple yet effective [36, 44]. Since the number of parts (*i.e.* k^2) is fixed as a hyper-parameter, it is easy to have a complete detection heatmap $z_{i,j,c}$ already computed for each region (i, j) of each class c (left of Fig. 2). Parts then only need to be optimized within corresponding maps.

The deformation of parts draws ideas from the original DPM [8]: it allows parts to slightly move around their reference positions (partitions of the initial regions), selects the optimal latent displacements, and pools values from selected locations. The pooled score $p_c^R(i, j)$ for part (i, j) and class c is a trade-off between maximizing the score on the feature map and minimizing the displacement (dx, dy) from the reference position (see Fig. 2):

$$p_c^R(i,j) = \max_{dx,dy} \left[\underset{(x,y)\in R_{i,j}}{\text{Pool}} \; z_{i,j,c}(x+dx, y+dy) - \lambda^{def} \left(dx^2 + dy^2\right) \right] \tag{1}$$

where λ^{def} represents the strength of the regularization (small deformations), and Pool is an average pooling as in [5], but any pooling function could be used instead. The deformation cost is here the squared distance of the displacement on the feature map, but other functions could be used equally. Implementation details can be found in Appendix A.1.

During training, deformations are optimized without part-level annotations. Displacement computed during the forward pass are stored and used to backpropagate gradients at

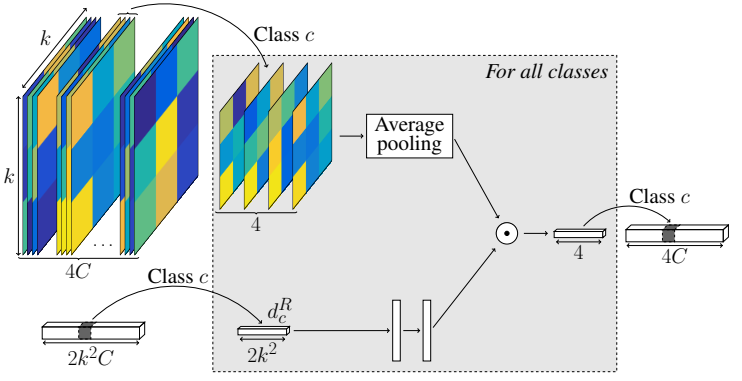


Figure 3: **Deformation-aware localization refinement.** Relocalizations of bounding boxes obtained by averaging pooled values from localization maps (upper path) do not benefit from deformable parts. To do so, displacements of parts are forwarded through two fully connected layers (lower path) and are element-wise multiplied with previous output to refine it, separately for each class. Localization is done with 4 values per class, following [13, 14].

the same locations. We further note that the deformations are computed for all parts and classes independently. However, no deformation is computed for the *background* class: they would not bring any relevant information as there is no discriminative element for this class. The same displacements of parts are used to pool values from the localization maps.

λ^{def} is directly linked to the magnitudes of the displacements of parts, and therefore to the deformations of RoIs too, by controlling the squared distance regularization (*i.e.* preference for small deformations). Increasing it puts a higher weight on the regularization and effectively reduces displacements of parts, but setting it too high prevents parts from moving and removes the benefits of our approach. It is noticeable this deformable part-based RoI pooling is a generalization of position-sensitive RoI pooling from [5]. Setting $\lambda^{def} = +\infty$ clamps dx and dy to 0, leading to the formulation of position-sensitive RoI pooling:

$$p_c^R(i,j) = \underset{(x,y)\in R_{i,j}}{\text{Pool}} \; z_{i,j,c}(x,y). \tag{2}$$

On the other hand, setting $\lambda^{def} = 0$ removes regularization and parts are then free to move. With λ^{def} too low, the results decrease, indicating that regularization is practically important. However the results appeared to be stable within a large range of values of λ^{def}.

3.3 Classification and localization predictions with deformable parts

Predictions are performed with two sibling branches for classification and relocalization of region proposals as is common practice [13]. The classification branch is simply composed of an average pooling followed by a SoftMax layer. This is the strategy employed in R-FCN [5], however the deformations introduced before (with deformable part-based RoI pooling) bring more invariance to transformations of objects and boost classification.

Regarding localization, we also use an average pooling to compute a first localization output from corresponding features. However, the configuration of parts (*i.e.* their positions

relative to each others) is obtained as a by-product of the alignment of parts performed before. It gives rich geometric information about the appearances of objects, *e.g.* their shapes or poses, that can be used to enhance localization accuracy.

To that end we introduce a new deformation-aware localization refinement module (see Fig. 3). For each region R, we extract the feature vector d_c^R of displacements (dx, dy) for all parts of class c (as shown on Fig. 2) and use it to refine previous output for the same class. d_c^R is forwarded through two fully connected layers and is then element-wise multiplied with the first values to yield the final localization output for this class. Since refinement is mainly geometric, it is done for all classes separately and parameters are shared between classes.

4 Experiments

4.1 Ablation study

Experimental setup. We perform this analysis with the fully convolutional backbone architecture ResNet-50 [17] and exploit the region proposals computed by AttractioNet [11, 12] released by the authors. We use $k \times k = 7 \times 7$ parts, as advised by the authors of R-FCN [5]. Setting of all others hyper-parameters can be found in Appendix B.1.

All experiments in this section are conducted on the PASCAL VOC 07+12 dataset [7]: training is done on the union of the 2007 and 2012 trainval sets and testing on the 2007 test set. In addition to the standard mAP@0.5 (*i.e.* PASCAL VOC style) metric, results are also reported with the mAP@0.75 and mAP@[0.5:0.05:0.95] (*i.e.* MS COCO style) metrics to thoroughly evaluate the effects of proposed improvements.

Comparison with R-FCN. Performances of our implementation of R-FCN [5] with the given setup are shown in the first row of Tab. 1. Adding the deformable part-based RoI pooling to R-FCN (second row of Tab. 1) improves mAP@0.5 by 2.1 points. Indeed, this metric is rather permissive so the localization does not need to be very accurate: we see that the gain on mAP@0.75 is much smaller. The improvements are therefore mainly due to a better recognition, thus validating the role of deformable parts. With the localization refinement module (third row of Tab. 1), the mAP@0.5 has only a small improvement, because localization accuracy is not a issue. However, it further improves mAP@0.75 by 2.1 points (*i.e.* 2.6 points with respects to R-FCN), validating the need for such a module. This confirms that aligning parts brings geometric information useful for localization.

Model	Deformations	Localization refinement	mAP@ 0.5	mAP@ 0.75	mAP@ [0.5:0.95]
R-FCN			73.7	38.3	39.8
	✓		75.8	38.8	40.4
DP-FCN	✓	✓	**76.1**	**40.9**	**41.3**

Table 1: **Ablation study of DP-FCN** on PASCAL VOC 2007 test in average precision (%). Without deformable part-based RoI pooling nor localization refinement module, it is equivalent to R-FCN (the reported results are those of our implementation with the given setup).

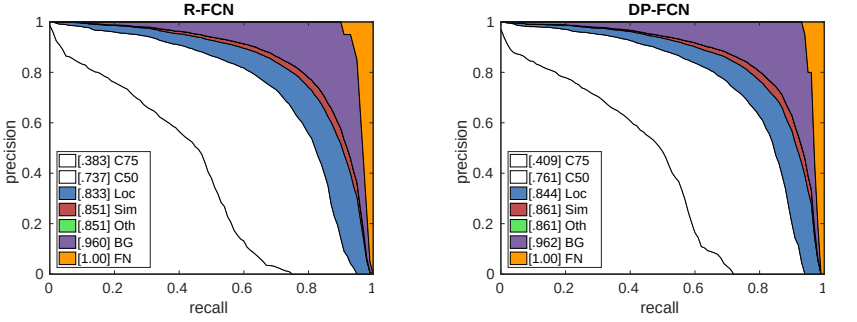


Figure 4: **Precision-recall curves for R-FCN (left) and DP-FCN (right).** Detailed analysis of false positives on unseen VOC07 test images averaged over all categories.

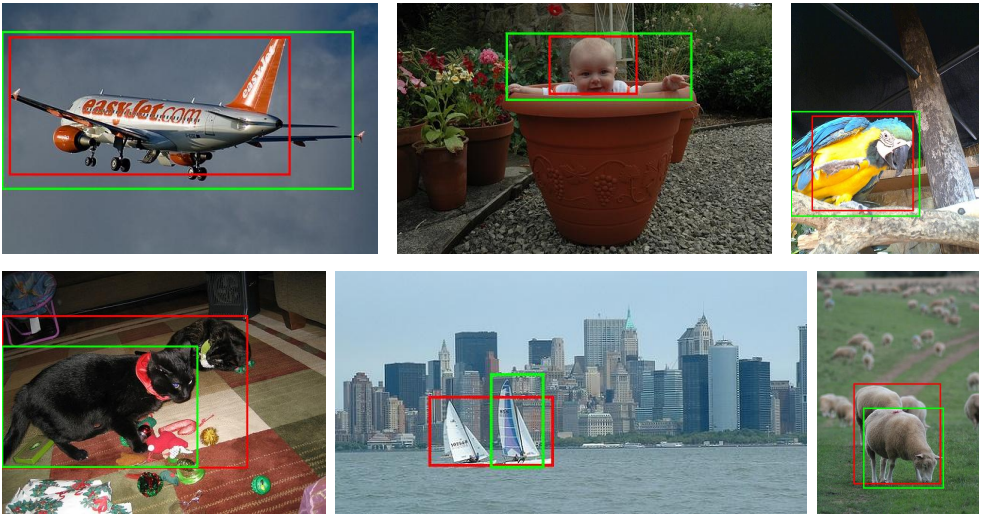


Figure 5: **Example detections of R-FCN (red) and DP-FCN (green).** DP-FCN tightly fits objects (first row) and separates close instances (second row) better than R-FCN.

Detailed breakdowns of false positives are provided in Fig. 4 for R-FCN and DP-FCN.[1] We see that the biggest gain comes from reduced localization errors (C75 and C50 metrics), and the corresponding curves are higher for DP-FCN. Ignoring those errors, recognition accuracy is consistently around 1 point better (Loc and Oth metrics). However, both models roughly keep the same number of false negatives (BG metric).

Examples of detection outputs are illustrated in Fig. 5 to visually evaluate proposed improvements. It appears that R-FCN can more easily miss extremal parts of objects (see first row, *e.g.* the right wing of the plane) and that DP-FCN is better at separating close instances (see second row, *e.g.* the two sheep one behind the other), thanks to deformable parts.

Interpretation of parts. As in the original DPM [8], the semantics of parts is not explicit in our model. Part positions are instead automatically learned to optimize detection perfor-

[1] See http://mscoco.org/dataset/#detections-eval for full details of metrics.

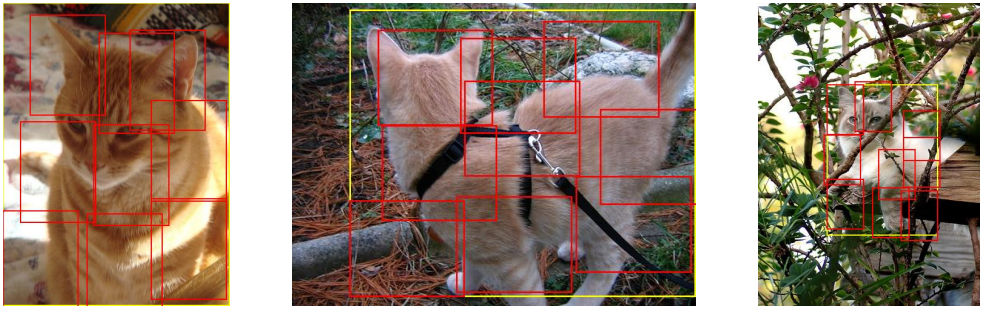


Figure 6: **Examples of deformations of parts.** Initial region proposals are shown in yellow and deformed parts in red. Only 3×3 parts are displayed for clarity.

mances, in a weakly supervised manner. Therefore the interpretation in terms of semantic parts is not systematic, especially because our division of regions into parts is finer than in DPM, leading to smaller part areas. Some deformed parts are displayed on Fig. 6 with a 3×3 part division for easier visualization. It is noticeable that DP-FCN is able to better fit to objects with deformable parts than with simple bounding boxes.

Network architecture. We compare DP-FCN with several FCN backbone architectures in Tab. 2, in particular the 50- and 101-layer versions of ResNet [17], Wide ResNet [40] and ResNeXt [38]. We see that the detection mAP of DP-FCN can be significantly increased by using better networks. ResNeXt-101 (64x4d) gives the best results among the tested ones, with large improvements in all metrics, despite not using dilated convolutions.

FCN architecture for DP-FCN	mAP@0.5	mAP@0.75	mAP@[0.5:0.95]
ResNet-50 [17]	76.1	40.9	41.3
ResNeXt-50 (32x4d) [38]*	76.3	40.8	41.4
Wide ResNet-50-2 [40]	77.9	43.3	42.9
ResNet-101 [17]	78.1	44.2	43.6
ResNeXt-101 (32x4d) [38]*	78.6	45.2	44.4
ResNeXt-101 (64x4d) [38]*	**79.5**	**47.8**	**45.7**

Table 2: **Comparison of DP-FCN with different FCN architectures** on PASCAL VOC 2007 test in average precision (%). Entries marked with * do not use dilated convolutions.

4.2 PASCAL VOC results

Experimental setup. We bring the following improvements to the setup of Section 4.1, the details of which are in Appendix B.2: we use ResNeXt-101 (64x4d) [38] and increase the number of iterations. We include common tricks: color data augmentations [19], bounding box voting [10], and averaging of detections between original and flipped images [2, 41]. We set the relative weight of the multi-task (classification/localization) loss [13] to 7 and enlarge input boxes by a factor 1.3 to include some context.

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
FRCN [8]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
HyperNet [23]	76.3	77.4	83.3	75.0	69.1	62.4	83.1	87.4	87.4	57.1	79.8	71.4	85.1	85.1	80.0	79.1	51.2	79.1	75.7	80.9	76.5
Faster R-CNN [18]	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0
SSD [17]	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
MR-CNN [10]	78.2	80.3	84.1	78.5	70.8	68.5	88.0	85.9	87.8	60.3	85.2	73.7	87.2	86.5	85.0	76.4	48.5	76.3	75.5	85.0	81.0
LocNet [7]	78.4	80.4	85.5	77.6	72.9	62.2	86.8	87.5	88.6	61.3	86.0	73.9	86.1	87.0	82.6	79.1	51.7	79.4	75.2	86.6	77.7
FRCN OHEM [32]	78.9	80.6	85.7	79.8	69.9	60.8	88.3	87.9	89.6	59.7	85.1	**76.5**	87.1	87.3	82.4	78.8	53.7	80.5	78.7	84.5	80.7
ION [1]	79.4	82.5	86.2	79.9	71.3	67.2	88.6	87.5	88.7	60.8	84.7	72.3	87.6	87.7	83.6	82.1	53.8	81.9	74.9	85.8	81.2
R-FCN [6]	80.5	79.9	87.2	81.5	72.0	69.8	86.8	88.5	89.8	**67.0**	**88.1**	74.5	89.8	90.6	79.9	81.2	53.7	81.8	**81.5**	85.9	79.9
DP-FCN [ours]	**83.1**	**89.8**	**88.6**	**85.2**	**73.9**	**74.7**	**92.1**	**90.4**	**94.4**	58.3	84.9	75.2	**93.4**	**93.1**	**87.4**	**85.9**	**53.9**	**85.3**	80.0	**90.4**	**85.9**

Table 3: **Detailed detection results on PASCAL VOC 2007 test** in average precision (%). For fair comparisons, the table only includes methods trained on PASCAL VOC 07+12.

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
FRCN [8]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
HyperNet [23]	71.4	84.2	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7
Faster R-CNN [18]	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
SSD [17]	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
FRCN OHEM [32]	76.3	86.3	**85.0**	77.0	60.9	59.3	81.9	81.1	91.9	55.8	80.6	**63.0**	90.8	85.1	85.3	80.7	54.9	78.3	70.8	82.8	74.9
ION [1]	76.4	88.0	84.6	77.7	63.7	63.6	80.8	80.8	90.9	55.5	81.9	60.9	89.1	84.9	84.2	83.9	53.2	79.8	67.4	84.4	72.9
R-FCN [6]	77.6	86.9	83.4	81.5	63.8	62.4	81.6	81.1	93.1	58.0	83.8	60.8	**92.7**	86.0	84.6	84.4	59.0	80.8	68.6	86.1	72.9
DP-FCN [ours][2]	**80.9**	**89.3**	84.2	**85.4**	**74.4**	**70.0**	**84.0**	**86.2**	**93.9**	**62.9**	**85.1**	62.7	**92.7**	**87.4**	**86.0**	**86.8**	**61.3**	**85.1**	**74.8**	**88.2**	**78.5**

Table 4: **Detailed detection results on PASCAL VOC 2012 test** in average precision (%). For fair comparisons, the table only includes methods trained on PASCAL VOC 07++12.

PASCAL VOC 2007 and 2012. Results of DP-FCN along with those of recent methods are reported in Tab. 3 for VOC 2007 and in Tab. 4 for VOC 2012. For fair comparisons we only report results of methods trained on VOC07+12 and VOC07++12 respectively, but using additional data, *e.g.* COCO images, usually improves results [5, 17]. DP-FCN achieves 83.1% and 80.9% on these two datasets, yielding large gaps with all competing methods. In particular, DP-FCN outperforms R-FCN [6], the work closest to ours, by significant margins (2.6% and 3.3% respectively). We note that these results could be further improved with additional common enhancements, *e.g.* multi-scale training and testing [16] or OHEM [32].

5 Conclusion

In this paper, we propose DP-FCN, a new deep model for object detection. While traditional region-based detectors use generic bounding boxes to extract features from, DP-FCN is more flexible and focuses on discriminative elements to align them. It learns a part-based representation of objects in an efficient way with a natural integration into FCNs and without any additional annotations during training. This improves both recognition by building invariance to local transformations, and localization thanks to a dedicated module explicitly leveraging computed positions of parts to refine predictions with geometric information. Experimental validation shows significant gains on several common metrics. As a future work, we will test our model on a larger-scale dataset, such as MS COCO [23].

[2] http://host.robots.ox.ac.uk:8080/anonymous/QNUYVS.html

A Implementation details

A.1 Deformable part-based RoI pooling layer

We normalize the displacements (dx, dy) by the widths and heights of parts to make the layer invariant to the scales of the images. We also normalize the classification feature maps before forwarding them to deformable part-based RoI pooling layer to ensure classification and regularization terms are comparable. We do this by L_2-normalizing at each spatial location the block of $C+1$ maps for each part separately, *i.e.* replacing z from Eq. (1) with

$$\bar{z}_{i,j,c}(x,y) = \frac{z_{i,j,c}(x,y)}{\sqrt{\sum_{c'} z_{i,j,c'}(x,y)^2}}. \tag{3}$$

Optimization of (dx, dy) is performed by brute force in limited ranges and not whole images. With λ^{def} (Eq. (1)) not too small, the regularization effectively restricts values of the displacements, leaving the results of pooling unchanged. In all experiments, we use $\lambda^{def} = 0.3$.

A.2 Deformation-aware localization refinement

The localization module is applied for each class separately and takes the normalized displacements d_c^R of a class as input, of size $2k^2$ (*i.e.* a 2D displacement for each part). It is composed of two fully connected layers with a ReLU between them. The size of the first layer is set to 256 in all our experiments. The output from average pooling (upper path in Fig. 3) is the main outcome and is obtained from the visual features only without considering deformations. The one from the fully connected layers (lower path in Fig. 3) encodes the positions of parts, and is merged with the first with an element-wise product (both are of size 4 for each class) to adjust it accordingly to the exact locations where it was computed.

B Experimental setups

B.1 Ablation study

We use the fully convolutional backbone architecture ResNet-50 [17] whose model pre-trained on ImageNet is freely available. The network is trained with SGD for 60,000 iterations with a learning rate of $5 \cdot 10^{-4}$ and for 20,000 further iterations with $5 \cdot 10^{-5}$. The momentum parameter is set to 0.9 and the weight decay to 10^{-4}. Each mini-batch is composed of 64 regions from a single image at the scale of 600 px, selected according to Fast R-CNN [13]. Horizontal flipping of images with probability 0.5 is used as data augmentation. We exploit the region proposals computed by AttractioNet [11, 12], released by the authors. The top 2,000 regions are used for learning and the top 300 are evaluated during inference. We use $k \times k = 7 \times 7$ parts, as advised by the authors of R-FCN [5]. As is common practice, detections are post-processed with NMS.

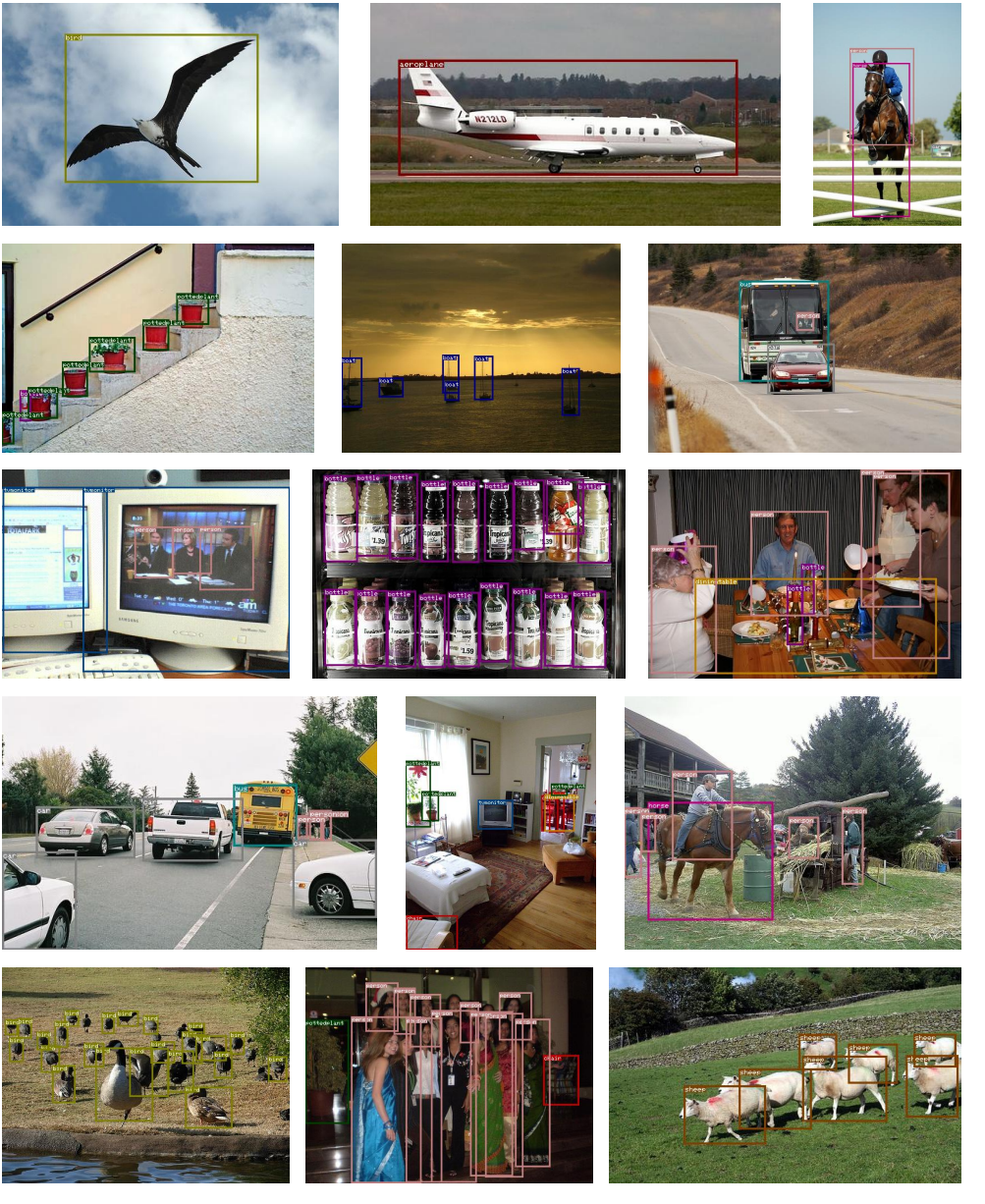
B.2 PASCAL VOC results

Changes with respect to the previous setup include replacing ResNet-50 by ResNeXt-101 (64x4d) [38], increasing the number of iterations to 120,000 and 160,000 with the same

learning rates, using 2 images per mini-batch with the same number of regions per image. We also include common tricks as described in the main paper.

C Examples of detections with DP-FCN

Below are some example detections (using VOC color code) on unseen VOC 2007 test images, from the final DP-FCN model trained on VOC 07+12 data (Section 4.2).



References

[1] Hossein Azizpour and Ivan Laptev. Object detection using strongly-supervised deformable part models. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 836–849, 2012.

[2] Sean Bell, Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[4] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 534–549, 2016.

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[6] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. WILDCAT: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Mark Everingham, Ali Eslami, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2015.

[8] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9):1627–1645, 2010.

[9] Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3294–3301, 2013.

[10] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware CNN model. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1134–1142, 2015.

[11] Spyros Gidaris and Nikos Komodakis. Attend refine repeat: Active box proposal generation via in-out localization. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[12] Spyros Gidaris and Nikos Komodakis. LocNet: Improving localization accuracy for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[15] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. Deformable part models are convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 437–446, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9):1904–1916, 2015.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. HyperNet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[20] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[21] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] Di Lin, Xiaoyong Shen, Cewu Lu, and Jiaya Jia. Deep LAC: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1666–1674, 2015.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott Reed. SSD: Single shot multibox detector. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2016.

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[27] Patrick Ott and Mark Everingham. Shared parts for deformable part-based models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1513–1520, 2011.

[28] Pedro Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 75–91, 2016.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[31] Pierre-André Savalle, Stavros Tsogkas, George Papandreou, and Iasonas Kokkinos. Deformable part models with CNN features. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV), Parts and Attributes Workshop*, 2014.

[32] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[33] Ronan Sicre, Yannis Avrithis, Ewa Kijak, and Frédéric Jurie. Unsupervised part learning for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[34] Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1143–1151, 2015.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[36] Li Wan, David Eigen, and Rob Fergus. End-to-end integration of a convolution network, deformable parts model and non-maximum suppression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 851–859, 2015.

[37] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Joint object and part segmentation using deep learned potentials. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1573–1581, 2015.

[38] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[39] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[41] Sergey Zagoruyko, Adam Lerer, Tsung-Yi Lin, Pedro Pinheiro, Sam Gross, Soumith Chintala, and Piotr Dollar. A MultiPath network for object detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

[42] Han Zhang, Tao Xu, Mohamed Elhoseiny, Xiaolei Huang, Shaoting Zhang, Ahmed Elgammal, and Dimitris Metaxas. SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1143–1152, 2016.

[43] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based R-CNNs for fine-grained category detection. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 834–849, 2014.

[44] Long Zhu, Yuanhao Chen, Alan Yuille, and William Freeman. Latent hierarchical structural learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1062–1069, 2010.