Deformable Part-based Fully Convolutional Network for Object Detection – Supplementary

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

1 Implementation details

1.1 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

$$\bar{z}_{i,j,c}(x,y) = \frac{z_{i,j,c}(x,y)}{\sqrt{\sum_{c'} z_{i,j,c'}(x,y)^2}}. \tag{1}$$

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 $\lambda^{def} = 0.3$.

1.2 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.

2 Experimental setups

2.1 Ablation study

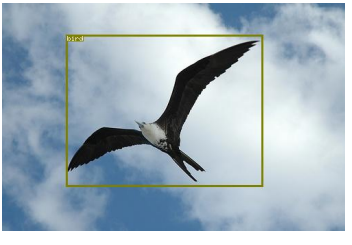
We use the fully convolutional backbone architecture ResNet-50 [17] whose model pre-trained on ImageNet is freely available. The network is trained with SGD for 60,000 iterations with a learning rate of $5 \cdot 10^{-4}$ and for 20,000 further iterations with $5 \cdot 10^{-5}$. The momentum parameter 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.

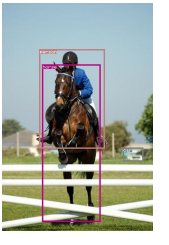
2.2 PASCAL VOC results

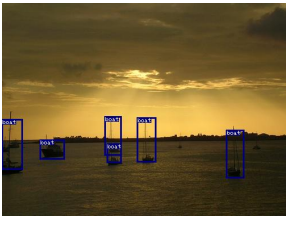
Changes with respect to the previous setup include replacing ResNet-50 by ResNeXt-101 (64x4d) [38], increasing the number of iterations to 120,000 and 160,000 with the same learning rates, using 2 images per mini-batch with the same number of regions per image. We also include common tricks as described in the main paper.

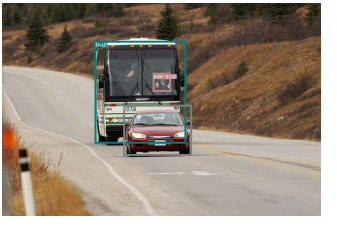
3 Examples of detections with DP-FCN

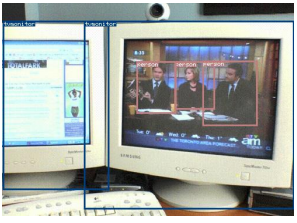
Below are some example detections (using VOC color code) on unseen VOC 2007 test images, from the final DP-FCN model trained on VOC 07+12 data (Section 4.2).

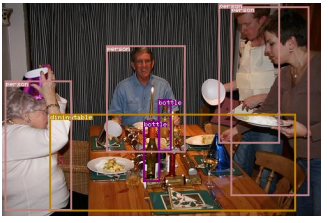
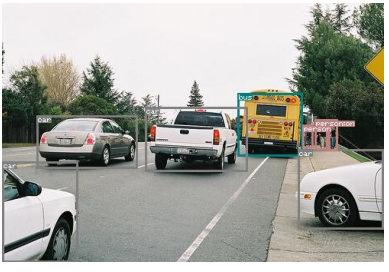
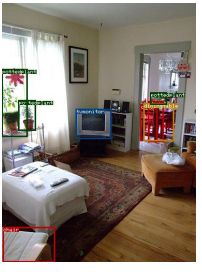


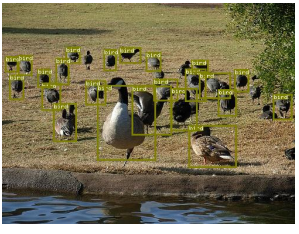
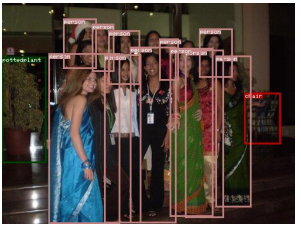
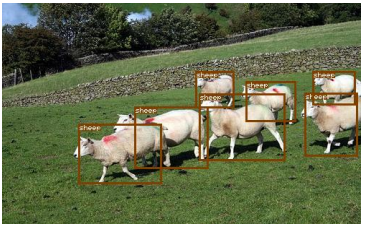




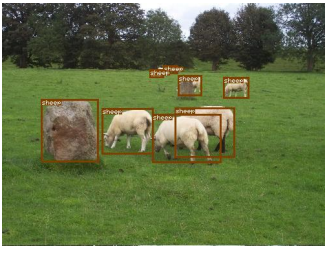
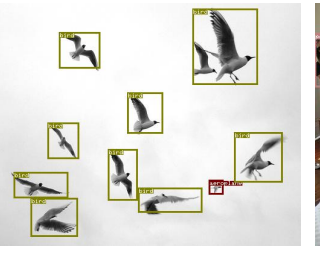







11.25.2006 10:35