Non-rectangular Part Discovery for Object Detection

Chunluan Zhou
czhou002@e.ntu.edu.sg
Junsong Yuan
jsyuan@ntu.edu.sg

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798

The deformable part-based model (DPM) is commonly used for object detection and many efforts have been made to improve the model. However, much less work has been done to discover parts for DPM. Most DPM-based methods adopt the greedy search approach proposed in [2] to initialize a predefined number of parts of rectangular shapes, which may not be optimal for some object categories. Moreover, object structures are not well exploited by the approach. In [4], a three-layer spatial pyramid structure is used to simplify the initialization of parts. An And-Or tree model [3] is proposed to select discriminative part configurations by a dynamic programming algorithm. Although the method can determine part sizes automatically, part shapes are still restricted to rectangles. To address the limitations of these methods, we propose a novel data-driven approach to discover non-rectangular parts by exploiting object structures. Figure 1 shows rectangular and non-rectangular parts obtained by the greedy search approach and our approach, respectively. Generally, the parts obtained by our approach can better cover object regions.

Figure 1: Rectangular parts vs. non-rectangular parts.

The DPM of an object category has several components representing different poses or orientations. Each component consists of a root which represents a whole object and a set of part filters which can move relatively to the root to capture structural deformations. As the training data only have bounding-box annotations specifying the image regions of training examples, the model is trained by first initializing the components and then learning model parameters in a latent structural SVM framework (See [2] for details). As the objective function used in the framework is not convex and as pointed out in [2] the training process is susceptible to local minima, it is necessary to select a good initialization of the components. In this paper, we focus on how to better initialize each component, especially its part filters.

Let M_c be the c-th component which has N_c part filters. The component M_c is defined by a $(2N_c+2)$-tuple $\beta_c = (\boldsymbol{F}_0, \boldsymbol{F}_1, \ldots, \boldsymbol{F}_{N_c}, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{N_c}, b)$, where $\boldsymbol{F}_0$ is the root filter, $\boldsymbol{d}_i \in \mathbb{R}^4$ is the deformation parameters of the part filter $\boldsymbol{F}_i$, and b is the bias term. Each filter $\boldsymbol{F}_i$ is an $H_i \times W_i$ array of n-dimensional weight vectors, where H_i and W_i are the height and width of $\boldsymbol{F}_i$, respectively. To initialize M_c, we first obtain the root filter $\boldsymbol{F}_0$ and then derive part filters from the root filter. The training examples are clustered into several groups each of which corresponds to one component. Let D_c be the set of object examples belonging to the c-th sub-category. $\boldsymbol{F}_0$ is obtained by training a linear SVM on the object examples in D_c and randomly sampled negative examples with each training example represented by histogram of oriented gradients (HOG) [1].

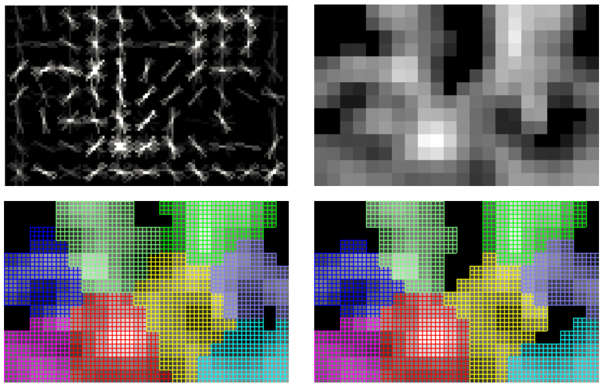

Figure 2: The process of our part discovery approach.

After F_0 is obtained, we find N_c part filters that have good matching regions on object examples in D_c and are consistent with these examples in terms of object structure. First, we double the size of the root filter $\boldsymbol{F}_0$ by interpolation, as in [2], to capture finer details. The enlarged root filter, denoted by $\boldsymbol{F}'_0$, is represented by a $2H_0 \times 2W_0$ array of cells C_k for $1 \le k \le 2H_0 \times 2W_0$, where each cell C_k corresponds to a n-dimensional weight vector in $\boldsymbol{F}'_0$. Then, from $\boldsymbol{F}'_0$, we obtain a configuration of N_c connected part filters, $\Lambda = \{\boldsymbol{F}_i | 1 \le i \le N_c\}$, which satisfies the following overlapping constraint:

$$O(\boldsymbol{F}_i, \boldsymbol{F}_j) = \frac{Area(\boldsymbol{F}_i \cap \boldsymbol{F}_j)}{Area(\boldsymbol{F}_i \cup \boldsymbol{F}_j)} < \tau \quad \text{for } i \neq j, \tag{1}$$

where τ is an overlapping threshold. This constraint prevents any two part filters from overlapping largely. We measure the fitness of the part filter configuration Λ to object examples in D_c by

$$F(\Lambda) = S_R(\Lambda)^{\lambda} \times S_C(\Lambda), \tag{2}$$

where $S_R(\Lambda)$ is the average matching response of Λ over object examples in D_c, $S_C(\Lambda)$ reflects the structural consistency of Λ with these examples, and λ is a parameter used to balance $S_R(\Lambda)$ and $S_C(\Lambda)$. Our goal is to find a feasible part-filter configuration Λ that maximizes $F(\Lambda)$. We refer readers to the paper for details on how $S_R(\Lambda)$ and $S_C(\Lambda)$ are defined and how the objective function is optimized. Figure 2 illustrates the process of our part discovery approach.

We test our approach on Pascal VOC2007 and VOC2010 datasets. Overall, our approach outperforms DPM for 19 and 17 out of 20 object categories in these two datsets respectively, which demonstrates the advantage of the discovered non-rectangular parts over the rectangular parts used in DPM. Implementation details and more experimental results are given in the paper.

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[2] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 2010.

[3] X. Song, T. Wu, Y. Jia, and S. Zhu. Discriminatively trained and-or models for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[4] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *International Conference on Computer Vision (ICCV)*, 2010.