Mixture Models for Object Detection

Xiaoqin Kuang, Nong Sang(✉), Feifei Chen, Changxin Gao, and Runmin Wang

Science and Technology on Multi-spectral Information Processing Laboratory,
School of Automation, Huazhong University of Science and Technology, Wuhan, China
{kxqkuang,nsang,ffchen,cgao,runminwang}@hust.edu.cn

Abstract. In this paper, we propose an approach based on mixture of multiple components and mid-level part models for object detection in natural scenes. It is difficult to represent an object category with a monolithic model as the intra-variance in the category. To solve this, we use multi-component models and part models to describe the global variation and local deformation respectively. We obtain multi-components by clustering to form visual similar object group and training discriminant model for each one. The mid-level part models are learned automatically. We apply max-pooling to generate the feature vector using all part models and then train the SVM classifier based on these feature vectors. When detecting in image, we first achieve object candidates using multi-component models, and then the performance is refined by using part models and SVM classifier. Experiments on standard benchmarks demonstrate this coarse-to-fine detection system performs competitively.

Keywords: Object detection · Mixture model · Multiple components · Mid-level parts · Max-pooling

1 Introduction

Object detection in natural scenes has been an important topic in computer vision for decades. However, it is still a challenge as the object varies greatly within each category. The variation comes not only from the illuminations, viewpoint, deformation, structure, but also the intra-variation of the object subcategory. Thus local deformation and global variation should be considered seriously.

Local variation is represented by local features. There are many researches have been focus on these issues. Hough transform based method [1, 2, 3, 4, 5, 6] has been applied popularly in the field. They vote with elements as boundaries, contours or patches which are collected with low-level features by point or edge detecting, line matching. They are not distinctive to the given object class. The Implicit Shape Model [1] generates codebooks by clustering patches which is found around interest point. Plenty discovered patches contain less semantic information with little structure information.

Mid-level semantic concepts corresponding to attributes and parts to promote the object detection have become more and more popular in recent years. These mid-level semantic concepts are commonly obtained by manually annotation or clustering on low-level features. Paper [7] proposes a convenient method to label parts in which

H. Zha et al. (Eds.): CCCV 2015, Part I, CCIS 546, pp. 317–324, 2015.
DOI: 10.1007/978-3-662-48558-3_32

parts are discovered with partial correspondence by annotating important matching points between instances of a category, which still need human participate in. Annotation work is time-consuming and hard to do when facing more new categories. Discovering mid-level semantic parts automatically is important and challenging.

Inspired by the works of discovering parts automatically [8, 9, 10], we obtain parts by learning discriminative detectors with training images. We follow the idea [8] to generate a small group patches for training part detectors, which is generalizing and consistent in semantic as well. Selecting a subset of discriminative part detectors in the training procedure is an important issue.

The trained part detectors represent local identity and robust to some structure deformation, while the global appearance also affords key information to the object detection. In order to solve the intra-variation in object, multiple components are proposed. Exemplar-SVM [12] trained separate HOG [11] detector for each positive exemplar. But training with single exemplar would not be expected generalize enough. The method of multi-component models in [13] picks 'seed' object and aligned the rest object to the seed as the component. It needs keypoint and mask annotation when aligning. Felzenszwalb and Girshick etc. [14] propose a detection system of mixtures of multi-scale deformable part models allowing for small deformation and multiple postures. But the numbers of components and parts are predefined and not referring to the data.

We follow their work of mixture models, but our multi-components and part models are discovered during training. In detection, the candidate windows are generated with multi-component models, and then part detectors are used to refine the performance. Each candidate is evaluated by the part models with feature vector using max-pooling technique. The overview of the proposed approach is shown in Fig. 1. Our approach shows competitive performance on the open dataset.

The rest of the paper is structured as follows. In Sec. 2 and Sec. 3 we describe the details of generating multi-components and discovering object parts. The experiments and analysis are presented in Sec. 4. Finally Sec. 5 concludes this paper.

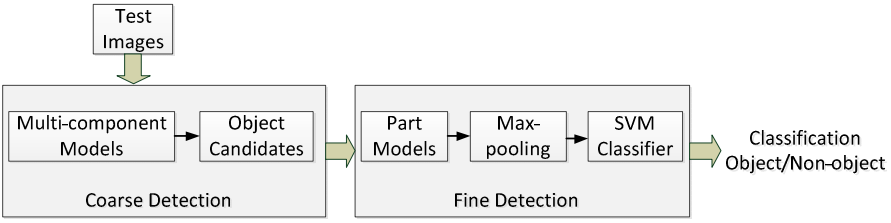

Fig. 1. Overview of our detection system with mixture models.

2 Multiple Components Generation

The challenge for object detection is the intra-variations induced by posture, viewpoint, subcategories, etc. The monolithic model is not enough to represent the object category of intra-variations. A natural strategy is to use multiple models to detect the

category. One way is to use models predefined by hand such as the different view or posture, but this is not convenient to choose artificially. The other way is to find models automatically.

We cluster using whitened HOG features. Whitened HOG features are considered to be better for clustering and classification [15], since the whitening removes the correlations and leaves the discriminative gradients. Firstly, we compute the covariance matrix Σ and mean feature μ_0 on HOG features with all background samples. Then we cluster the positive objects using whitened features which are transformed from the HOG feature x to $\hat{x} = \Sigma^{-1/2}(x - \mu_0)$.

We set a fixed number k in advance to the object cluster number according to the data complexity. If the data samples within the category have great variation in posture, structure, illumination, etc., we set k a large number. Conversely, we set k a small number when dada samples are simple and have not much variation. Components work as a guide to the coarse detection, so clustering is enough to generate generalizing and representative model which separate the variation among the category to different groups and forms similar visual groups. After clustering, we train LDA model to each of them since it has the similar performance with SVM but accelerates the computation [15]. The LDA model is a linear classifier over x with weights given by $\omega = \Sigma^{-1}(x_{mean} - \mu_0)$, and x_{mean} is the mean feature of the cluster. The covariance matrix Σ and mean feature μ_0 are both computed in advance with all background.

Finally, we measure the redundancy between a pair of components by their cosine similarity $\langle \omega_i/\|\omega_i\|, \omega_j/\|\omega_j\| \rangle$, redundant components are removed.

3 Discovering Mid-level Part Models

Candidate windows are generated to describe the global appearance of object, and then are sent to the part detectors for further measurement. In this section, we describe the procedure of discovering the mid-level parts. The flow is illustrated in Fig. 2.

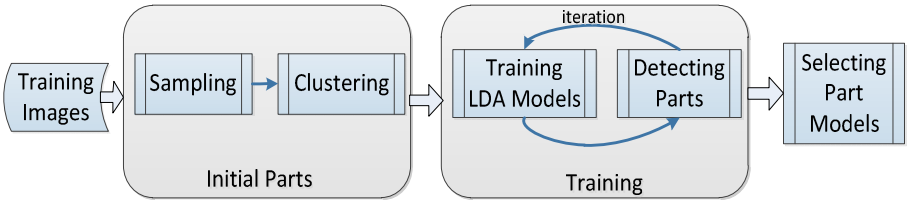

Fig. 2. Procedure of discovering of mid-level part models

3.1 Clustering

Given a set of training data, our goal is to learn a compact collection of part detectors for the category. Every training image is equally containing distinctive parts, so we

sample densely to get all possible sub-windows and collect hundreds of thousands of parts. Let $\mathcal{D}$ be object images, N be background images. Patches are represented by augmented HOG descriptors [14] with the same dimensions.

Clustering is applied to generate 'seed' for training the initial detectors. We run k-means to cluster patches. As mentioned in [8], the number of clusters is set quite high since they do not trust k-means to generalize well, so we set k quite high ($k = S/5$, where S is the number of patches sampled for clustering) similar to them to ensure the consistency of each cluster. Clusters with less than 4 patches are regarded as poor seeds and removed. We cluster the training samples with k-means using whitened HOG features same as the components generation.

3.2 Training

Naturally, clusters are probably rough and impure due to this unsupervised clustering. So the training scheme is followed which aims at collecting patches purer and more consistent.

We apply LDA to train detector for each cluster. In order to improve the consistency, we use the cross-validation to refine detectors. We divide $\mathcal{D}$ into two equal, non-overlapping subsets ($\mathcal{D}_1$, $\mathcal{D}_2$) as train-set and validation-set. Given the initial clusters obtained from train-set $\mathcal{D}_1$, we train the discriminative classifier for each of them, using patches in N as negative samples.

We run the trained detectors in validation-set $\mathcal{D}_2$ to discover the corresponding patches, and then new clusters are formed by the top m scores of detections. We set $m = 6$ to keep the purity of each cluster.

After this training and detection, we switch the role of $\mathcal{D}_1$ and $\mathcal{D}_2$ and repeat the process until convergence. During the iteration, the detections with small number of patches are eliminated since they occur rarely to characterize the appearance of object.

3.3 Part Selecting

The training procedure has produced numerous candidate detectors; the next task is to select a small number of the most discriminative ones. A desired detector should fire majority of the object category where it is learned. That is to say it should be representative to its class and occur frequently enough. What's more, an effective detector should be highly discriminative that it fires rarely in the natural world. To select a set of 'good' detectors, criteria need to be defined to measure the quality of detectors based on these two notions.

Concretely, for a part detector, we represent an object by a binary tuple $(\mathrm{s}_i; \mathrm{y}_i)$ where s_i is the highest detection score of patches within object i ($\mathrm{s}_i = \max_j \omega^T x(i)_j$, in which $x(\cdot)_j$ indicate one patch in the object region); y_i is the class label of this object. We set a high threshold τ and consider the patch with score $\mathrm{s}_i > \tau$ as a firing; thus the representativeness in positive samples is computed as

$$r = \frac{\sum_i 1(\mathrm{s}_i > \tau) \cdot \mathrm{s}_i}{|\mathcal{D}|}$$

where $1(\cdot)$ is the indicator function which equals 1 only if the corresponding logical expression is established and 0 otherwise, and $|\mathcal{D}|$ is the number of positive samples.

Following the conception in [8] we evaluate the discriminativeness of detectors as the ratio of number of firings on $\mathcal{D}$ to number of firings on $\mathcal{D} \cup \mathcal{N}$. We write it as

$$d = \frac{\sum_i 1(s_i > \tau) \cdot 1(y_i = y_0)}{\sum_i 1(s_i > \tau) \cdot 1(y_i = y_0) + \sum_i 1(s_i > \tau) \cdot 1(y_i = y_1)}$$

where y_0 is the label of the object category from which the detector is learnt, and y_1 is the label of non-object.

All detectors are ranked using a linear combination of the above two scores. Then the detectors with top scoring are retained. Fig. 3 gives some examples of selected part with high scores. Patches are densely sampled so redundancy exists among parts. Thus the final step is to remove redundant part detectors. We measure the redundancy between a pair of detectors by their cosine similarity, detectors that have similarity larger than 0.5 from the last ranking to first are removed.

Fig. 3. Some selected part detectors with top scores.

4 Experiments

Object Detection: Component models are applied to capture the entire object, so they are defined by coarse template which has lower resolution than part models. We expand the components by flipping the models to discover the mirrored object in images. Part detectors are used to generate the image-level descriptor for the detection candidates. The image vector is formed by summarizing the part scores of selected detectors using max-pooling. We use a 2-level spatial pyramid ($1 \times 1, 2 \times 2$ grids) to pooling; we stack the maximum detection score of each detector of each cell in the pyramid. Finally encodings of all the detectors are concatenated to form the feature vector of image representation. We train a linear-SVM using these obtained feature vector of object and background. Then each candidate is evaluated by the classifier. We follow up traditional object detection methods to search in images in a sliding-window manner with multiple components. The obtained candidates will be measured by part models. We generate feature vector for each candidate by max-pooling and then classify it as object or non-object by the learned SVM classifier.

Dataset: The UIUC Cars single-scale test set contains 170 images with 200 side views of cars of approximately the same scale. The multi-scale test set contains 108 images with 139 cars at multiple scales. The images are low contrast with some cars partially occluded and multiple objects would occur. The training set contains 550 training cars of size 100×40 and 500 negative training examples of the same size.

Implementation Details: As the training objects have the same scale of 100×40, we describe object with 13×5 cells by augmented HOG features of dimensionality $d = 31$. The object descriptor has 2015 dimensions after concatenating all cells. The part models have twice resolution of the component models, each HOG cell is 4×4 pixels. We obtain patches by sampling densely with 7×7 cells and the patch descriptor has 1519 dimensions. The sliding-window search for multiple scale test set detection is in multiple scale. We set number of levels in an octave $\lambda = 10$ as the scale in the feature pyramid. Fine sampling of scale space is important for high detection performance.

Results: We adhered to the experimental evaluation criteria based on bounding-box overlap as previous works. The detections are considered be true as its overlap area with the ground truth less than 0.5.

Using the multi-component models, we obtained recall at 99.5% and precision at 72.6% with for the UIUC-Single; and recall at 99.3% and precision at 50.2% for the UIUC- Multi. The recall is high but there existing a lot of false positive in the results.

We use the 154 learned part detectors to remove the false object and refine the performance. Finally, with the mixture models, we get the recall at 99.5%, precision at 99.5% and recall-precision EER at 99.5% for the single-scale test set; recall at 99.3%, precision at 96.5% and recall-precision EER at 99.3% for the multi-scale test set. Table 1 presents the results at recall-precision equal error rate compared to other methods.

Table 1. Performance of different methods on the two UIUC datasets at recall-precision equal error rate (EER)

Methods	ISM. No MDL	ISM +MDL	Hough Forest	HF Weaker supervision	Our approach
UIUC-Single	91%	97.5%	98.5%	94.4%	99.5%
UIUC-Multi	-	95%	98.6%	-	99.3%

In order to analyze the impact of the part models numbers, we select a set of different number of detectors with percentage from 10% to 100% (the detector number varies from 15 to 154). The performance changes are illustrated in Fig. 4 and Fig. 5. Despite some fluctuates existing, we can see that the mixture system still performs quite well compared to entire object models only.

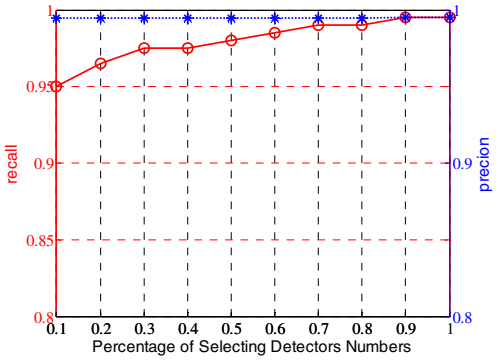

Fig. 4. Performance changes as the detectors numbers changes of UIUC-Single set

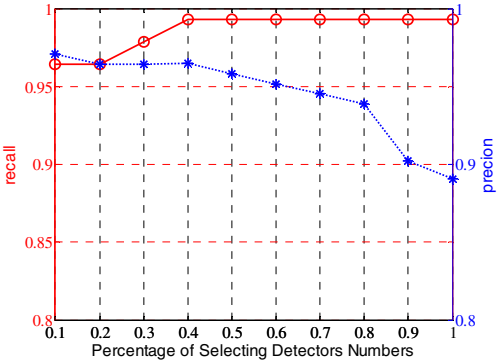

Fig. 5. Performance changes as the detectors numbers changes of UIUC-Multi set

For the UIUC-Single set, the recall gradually promotes when the detectors amount increasing, while the precision stays stable. For the UIUC-Multi set, the recall gradually promotes and tends stable at last, but the precision fluctuates and descends gradually when the amount increasing.

For a certain small amount of part detectors, the number increasing will promote the performance. However, for a certain great amount of part detectors, the detection performance will have some decrease as the number increasing. This is because the quality of detectors are sorted by measurement in descending order and the bottom ones have poor representativeness and descriminativeness which usually refer to unimportant part or the background. This further shows that part models can represent the object appearance well as if selecting a group of fine models.

5 Conclusion

We have presented a method of object detection. This mixture models of multi-components and part models can represent the global appearance of object, and also adapt to the intra-variation of the category. The key problem of generating distinctive

semantic parts is solved by the proposed measure criterion. We showed the ability of part models and multiple components for object detection. Our method achieves competitive performance on the dataset.

Acknowledgements. This research is supported by the Natural Science Foundation of China No.61401170 and No.61433007).

References

1. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. International Journal of Computer Vision **77**(1–3), 259–289 (2008)
2. Gall, J., Yao, A., Razavi, N., Van Gool, L.: Hough forests for object detection, tracking, and action recognitions. IEEE Transactions on Pattern Analysis and Machine Intelligence. **33**(11), 2188–2202 (2011)
3. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1038–1045. Miami, FL (2009)
4. Kontschieder, P., Riemenschneider, H., Donoser, M., Bischof, H.: Discriminative learning of contour fragments for object detection. In: BMVC, pp. 1–12 (2011)
5. Razavi, N., Gall, J., Van Gool, L.: Scalable multi-class object detection. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1505–1512. IEEE (2011)
6. Razavi, N., Gall, J., Kohli, P., van Gool, L.: Latent hough transform for object detection. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 312–325. Springer, Heidelberg (2012)
7. Maji, S., Shakhnarovich, G.: Part discovery from partial correspondence. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 931–938. Portland, OR (2013)
8. Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: European Conference on Computer Vision, pp. 73–86 (2012)
9. Juneja, M., Vedaldi, A., Jawahar, C.V., Zisserman, A.: Blocks that shout: distinctive parts for scene classification. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 923–930. Portland, OR (2013)
10. Endres, I., Shih, K.J., Jiaa, J., Hoiem, D.: Learning collections of part models for object recognition. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 939–946. Portland, OR (2013)
11. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection, In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. San Diego, CA, USA (2005)
12. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-SVMs for object detection and beyond, In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 89-96. Barcelona (2011)
13. Gu, C., Arbeláez, P., Lin, Y., Yu, K., Malik, J.: Multi-component models for object detection. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 445–458. Springer, Heidelberg (2012)
14. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(9), 1627–1645 (2010)
15. Hariharan, B., Malik, J., Ramanan, D.: Discriminative decorrelation for clustering and classification. In: European Conference on Computer Vision, pp. 459–472 (2012)