Object Detection by Learning Oriented Gradients

Jiajie Chen[1,2,3], Huicheng Zheng[1,2,3](✉), Na He[1,2,3],
Ziquan Luo[1,2,3], and Rui Zhu[1,2,3]

[1] School of Data and Computer Science, Sun Yat-sen University,
135 West Xingang Road, Guangzhou 510275, China
zhenghch@mail.sysu.edu.cn

[2] Key Laboratory of Machine Intelligence and Advanced Computing,
Ministry of Education, Sun Yat-sen University, 135 West Xingang Road,
Guangzhou 510275, China

[3] Guangdong Key Laboratory of Information Security Technology,
Sun Yat-sen University, 135 West Xingang Road, Guangzhou 510275, China

Abstract. This paper proposes a method of learning features corresponding to oriented gradients for efficient object detection. Instead of dividing a local patch into cells with fixed sizes and locations such as in the traditional HOG, we employ a data-driven method to learn the sizes and locations of cells. Firstly, oriented gradient patch-maps of a local patch are constructed according to the orientations. Secondly, rectangular cells of various sizes and locations are constructed in each patch-map to sum up the magnitudes of oriented gradients and produce candidate local features. The local features are then selected by using a boosting procedure. Finally, a local patch is represented by a feature vector in which each component corresponds to the sum of oriented gradients in a rectangular cell. An object detector is then trained over the local patches by using a higher-level boosted cascade structure. Extensive experimental results on public datasets verified the superiority of the proposed method to existing related methods in terms of both the training speed and the detection accuracy.

Keywords: Oriented gradients · Boosting · Object detection

1 Introduction

With its wide applications in computer vision, object detection has become one of the most studied problems for several decades. Developing a reliable object detector enables a vast range of applications such as video surveillance [3] and the practical deployments of autonomous and semiautonomous vehicles [2] and robotics [1]. It is also a key component of many other computer vision tasks, such as object tracking [11], object recognition [7], scene understanding [5], and augmented reality [6]. The fundamental goal of object detection is to detect the locations and categories of multiple object instances in the images efficiently.

Y. Zhao et al. (Eds.): ICIG 2017, Part II, LNCS 10667, pp. 86–96, 2017.
https://doi.org/10.1007/978-3-319-71589-6_8

Generally, object detection consists of two steps: (1) feature extraction and (2) classification. Object detection based on deep learning attracted wide interests and showed promising performance recently [20,21]. However, the training process of deep learning models is generally very time-consuming even with the support of GPUs and requires large training datasets. In the milestone work of Viola and Jones [8], a boosted cascade of simple features is proposed for efficient object detection. Since then, many researchers have made efforts to extend the approach. The impressive improvement has been made mainly via: (1) improving the boosted cascade structure [4,23,25], and (2) learning low-level features based on appearance models [12–14]. There are three representative low-level features constructed based on gradient information, i.e., Histogram of Oriented Gradients (HOG) [15], SIFT [9], and SURF [24]. All the descriptors adopt position-fixed histograms computed in local cells for representation of local patches.

The cascade-HOG framework [10] is a representative method for constructing weak classifiers based on HOG features in local patches. To construct the HOG features, it is necessary to compute magnitudes and orientations of image gradients. The histograms are generated by adding up the gradient information in small spatial regions (cells). In this way, local object appearance and shape can be generally characterized by the distribution of local intensity gradients or edge directions. Such descriptions are invariant to local geometric and photometric transformations. Nevertheless, the handcrafted features are not adaptive to complicated distributions in real-world applications. In the HOG, a local patch is evenly divided into 4 cells to construct the histograms separately [10,15,19]. Such a fixed construction of cells, however, may not cope well with variations in various object classes, which could limit the capability of object detectors trained over local patches. To address this issue, in this paper, instead of dividing a local patch into cells with fixed sizes and locations such as HOG, we propose a data-driven method to learn local features corresponding to oriented gradients, intending to better capture the appearance and shape of various objects.

The proposed feature learning chain is summarized in Fig. 1. Firstly, by computing the orientation of the gradients for each local patch, we create k oriented gradient maps for each local patch. These maps, namely patch-maps in this paper, have the same size as the local patch. We then construct rectangular cells of various sizes and locations in each patch-map. Within each cell, the magnitudes of oriented gradient are accumulated over the pixels of cell as one candidate feature. Secondly, we use a boosting procedure to learn the local features, which selects a few features corresponding to dominant oriented gradients for each patch-map. The selected features of each patch-map are concatenated to form the descriptor of local patches. Finally, we train object detector over these local patches with a higher-level boosted cascade, which replaces the two conflicted criteria (false-positive rate and hit rate) with a single criterion AUC (area under the curve) for convergence test. Experiments on various public datasets verified that the proposed method obtained better performance than existing related methods both in terms of the training speed and the detection accuracy.

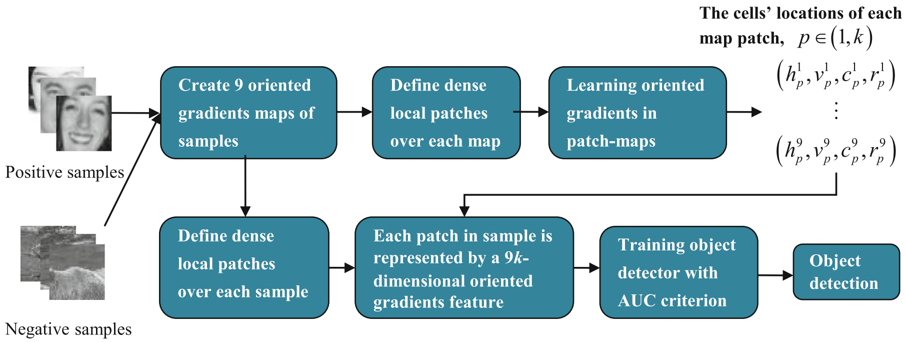
The cells' locations of each map patch, $p \in (1,k)$
Create 9 oriented gradients maps of samples
Define dense local patches over each map
Learning oriented gradients in patch-maps
$(h_p^1, v_p^1, c_p^1, r_p^1)$
$\vdots$
$(h_p^9, v_p^9, c_p^9, r_p^9)$
Positive samples
Negative samples
Define dense local patches over each sample
Each patch in sample is represented by a 9k-dimensional oriented gradients feature
Training object detector with AUC criterion
Object detection

Fig. 1. An overview of the proposed method for oriented gradient learning and object detection.

The rest of this paper is organized as follows. Section 2 describes the process of learning local features corresponding to oriented gradients in detail, and constructs the corresponding object detector. Experimental analysis is presented in Sect. 3. This paper is finally concluded by Sect. 4.

2 The Proposed Method

2.1 Learning Oriented Gradients

We collect positive and negative samples from the training set, and compute gradients of the sample images in advance.

In general, we define $G_x(x, y)$ as the horizontal gradient of a pixel at (x, y) by using the filter kernel $[-1, 0, 1]$, and $G_y(x, y)$ as the vertical gradient by the filter kernel $[-1, 0, 1]^T$. The magnitude $G(x, y)$ and orientation $\alpha(x, y)$ of the gradient are computed as follows.

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}, \tag{1}$$

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right), \tag{2}$$

The orientations of gradients ranging from 0° to 360° are evenly divided into k bins, k is set as 9 here. We generate an oriented gradient map with the same size as the input sample for each bin. If the gradient orientation of a pixel at (x, y) belongs to the i-th bin, we set the value at the same location in the i-th map to be the gradient magnitude of the pixel. The same locations in the other maps are then filled with 0.

We define patches of various sizes for reliable detection. For instance, given a template with 40×40 pixels, the patch sizes are set as 16×16 pixels, 16×32 pixels, 32×16 pixels, and 32×32 pixels. A window with a stride of 4 pixels slides over the oriented gradient maps to extract the local patches. We obtain a patch-map set for each oriented gradient map. The patch-map set is utilized for learning local features corresponding to oriented gradients, in which positive patch-maps are generated from the oriented gradient maps of positive samples and negative patch-maps are from negative samples. We take rectangular cells of various locations and sizes in a patch-map, and accumulate all gradients in a rectangular cell as one candidate feature. One weak classifier is built over each rectangular cell in parallel from the patch-map set. The decision tree is chosen as the model for weak classifiers for convenience.

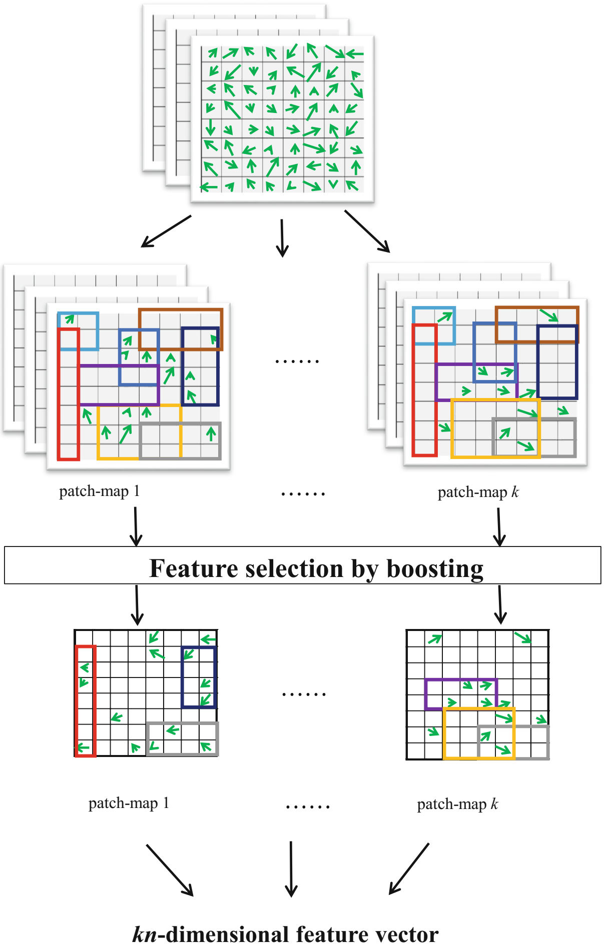
patch-map 1
......
patch-map *k*
Feature selection by boosting
patch-map 1
......
patch-map *k*
kn-dimensional feature vector

Fig. 2. The extraction process of oriented gradient features for a local patch. The arrow direction defines the orientation of gradients of a pixel, whereas the arrow length denotes the magnitude. There are k patch-maps corresponding to k orientation bins, where each one yields n features by summarizing gradients in n rectangular cells (the small rectangular regions in patch-maps). Totally kn oriented gradient features are generated for each local patch.

The AdaBoost learning algorithm is used to select a small number of weak classifiers, which can also be regarded as a process of rectangular cell selection. The boosting procedure is illustrated in Algorithm 1. We preserve the first n rectangular cells selected by boosting, where $n = 10$ according to our experiments. $(h_p^q, v_p^q, c_p^q, r_p^q)$ is recorded as the location of the rectangular cell, where (h_p^q, v_p^q) represents the coordinates of the upper left corner of the cell, c_p^q and r_p^q represent its width and height, $p = 1 : n$, $q = 1 : k$. Each local patch in the training sample is represented by a kn-dimensional feature vector where each component corresponds to the sum of oriented gradients in a selected rectangular cell. The process is illustrated in Fig. 2.

Algorithm 1. Learning oriented gradients with boosting.

```
Input: Given a patch-map set: ${(x_i, y_i)}_{i=1}^N$, where $N$ is the number of patch-maps
    in the set, $x_i$ is the $i$-th patch-map and $y_i$ is the label of $x_i$, $y_i = 1$ for a positive
    patch-map, $y_i = -1$ for a negative patch-map. We define $x_i^j$ as the feature of the
    $j$-th rectangular cell, $j = 1 : J$, where $J$ is the number of possible rectangular cells.
1: Initialize: Initial weights for positive and negative patch-maps: $w_{1,i} = \frac{1}{N}$, $i = 1 : N$.
2: Boosting: for $t = 1 : T$
3: Train a decision tree $s_j$ for the $j$-th rectangular cell. The error $\varepsilon_j$ is evaluated with
   respect to $w_t$: $\varepsilon_j = \sum_{i=1}^N w_{t,i}\delta(y_i \neq s_j(x_i))$, where $\delta(\cdot)$ is an indicator function,
   which outputs 1 if the argument is true and 0 otherwise.
4: Choose the classifier $\hat{s}_t$ with the lowest error $\hat{\varepsilon}_t$, and obtain the location
   $(h_t, v_t, c_t, r_t)$ of the corresponding rectangular cell.
5: Update weight $w_{t+1,i} = w_{t,i} \exp(-\alpha_t y_i \hat{s}_t(x_i))$, $\alpha_t = \log(\frac{(1-\hat{\varepsilon}_t)}{\hat{\varepsilon}_t})$.
6: Normalize the weights $w_{t+1}$ as a probability distribution.
Output: Output locations $(h_t, v_t, c_t, r_t)_{t=1}^T$ of the selected rectangular cells.
```

2.2 Object Detector Training

Each local patch in the training samples is represented by a kn-dimensional feature vector based on oriented gradients. Inspired by [24], for the object detector, we further build a weak classifier over each local patch by using logistic regression as it has probabilistic output. AUC is adopted as the single criteria for convergence test during the boosted cascade learning, which helps to accelerate the training speed.

Given an oriented gradient feature vector $\mathbf{x}$ of a local patch, which is a kn-dimensional feature vector, the classifier based on logistic regression is defined as follows:

$$g(\mathbf{x}) = \frac{1}{1 + \exp(-y(\mathbf{w}^T\mathbf{x} + b))}, \tag{3}$$

where $y \in \{-1, 1\}$ is the label of the local patch, $\mathbf{w}$ is a kn-dimensional weight vector and b is a bias term. We solve the following unconstrained optimization problem to obtain the parameters by using Liblinear [22],

$$\min_{\mathbf{w},b} \ C\sum_{i=1}^{L} \log\left(1+\exp(-y_i(\mathbf{w}^T\mathbf{x}_i+b))\right)+\|\mathbf{w}\|_1, \qquad (4)$$

where $C > 0$ is a penalty parameter, $\|\cdot\|_1$ denotes the l_1-norm, L is the number of training samples, $\mathbf{x}_i$ is a kn-dimensional feature vector of a local patch on the i-th training sample, y_i is the corresponding label of the local patch.

We implement the boosted cascade framework to train the object detector. Each stage of the cascade is a boosted learning procedure. In the r-th boosting round, we build a logistic regression model $g(\mathbf{x})$ for each local patch in parallel from the training set. Assume that there are M local patches in a sample. Then M logistic regression classifiers $\{g_m(\mathbf{x})\}_{m=1}^{M}$ would be created. $G^{r-1}(\mathbf{x})+g_m(\mathbf{x})$ is tested on all training samples to get an AUC score , where $G^{r-1}(\mathbf{x})$ is a combined classifier of previous $r-1$ rounds. We seek the classifier $g_r(\mathbf{x})$ which produces the highest AUC score. Gentle AdaBoost is adopted to combine the weak classifiers at the end of each boosting round.

The decision threshold θ of the strong classifier per stage is determined by searching on the ROC curve to find the point (d, f) such that the hit rate $d = d_{min}$, where d_{min} is the minimal hit rate for each stage. The value f is then the false positive rate (FPR) at the current stage. Therefore, FPR is adaptive across different stages, usually with values much smaller than 0.5. It means that the overall FPR can reach the overall goal of the cascade quickly. As a result, the cascade of stages tends to be short.

2.3 Object Detection

The trained object detector works on a fixed template size, but objects in images may have various sizes in practice. To guarantee effective detection of these objects, we build an image pyramid by scaling the input image.

The object detector scans across all scales in the pyramid to find objects. Considering that the detector may not be sensitive to small changes of scales, multiple detections may appear around a candidate object. We merge these duplicated detections with a simple strategy. All detections in an image are partitioned into disjoint subsets at first. Specifically, detections are in the same subset if their bounding boxes overlap. Each subset generates a final bounding box as detection, whose corners are the averages of the corresponding corners of all detections in the subset.

3 Experimental Results

The proposed approach is evaluated experimentally on three public datasets: UMass FDDB, PASCAL VOC 2007, and PASCAL VOC 2005. The experiments were all carried out with an Intel Xeon E5-2609 v4@1.70 GHz CPU.

3.1 Experiments on the FDDB Dataset

The FDDB dataset contains 2845 images with a total of 5771 faces that may be subject to substantial occlusion, blur, or pose variations. Five face detectors are separately trained for various views: the frontal view, the left/right half-profile views, and the left/right full-profile views.

For positive training samples, we collected 14000 faces from the frontal view, 8000 faces from the half-profile views, and 5000 faces from the full-profile views, which were all resized to 40 × 40 pixels. We also collected about 8000 images without faces, which were scanned by using sliding windows to construct negative training samples. The training procedure only took about 2 h to converge at the 6th stage, while the Haar training module provided by OpenCV took more than 2 days to finish training with the same dataset and CPU.

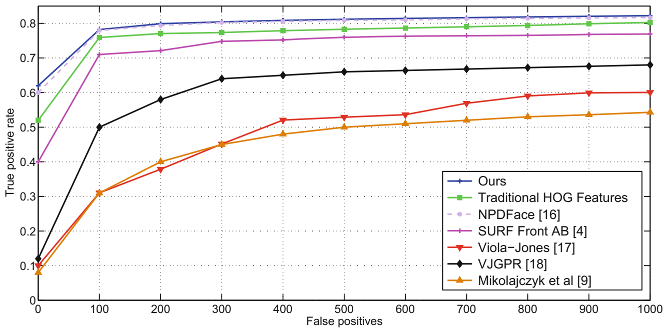

Fig. 3. ROC curves of various methods obtained on the FDDB dataset.

To demonstrate the advantage of the proposed features based on learning oriented gradients in local patches, we replaced the proposed feature vector with the traditional HOG in the boosted cascade detector. Figure 3 shows the ROC curves of various methods including ours on the FDDB dataset [4,9,16–18]. The proposed approach outperforms most of existing methods significantly. Specifically, the proposed features show clear improvement over the common HOG features. The proposed detector also slightly surpasses NPDFace specifically designed for face detection [16].

Figure 4 shows some examples of the detected faces by using the proposed features learned from oriented gradients in local patches in comparison to those by using traditional HOG features. The proposed method can successfully detect faces subject to various poses, occlusion, and illumination conditions, while the method based on the traditional HOG features failed to detect some partly occluded or blurred faces, and produced a number of false detections. Traditional HOG method with fixed cells in local patch can not capture the appearance and shape precisely. When applied to detect some partly occluded or blurred faces, HOG performs not so well. The results verified the superiority of the proposed features to the traditional HOG features in terms of face detection.

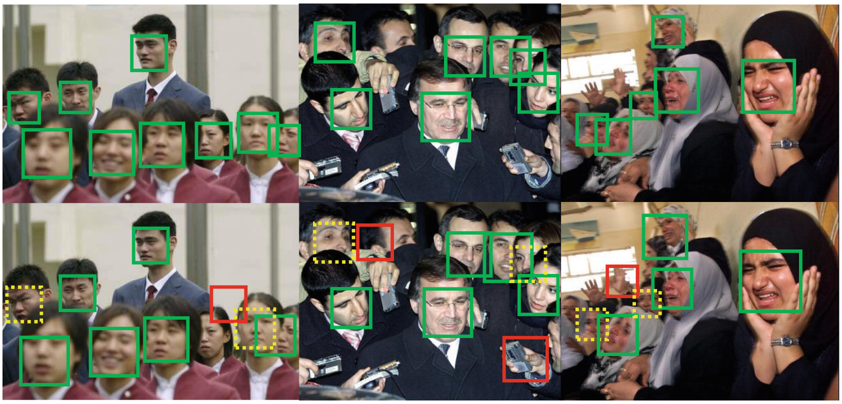

Fig. 4. Examples of face detection on the FDDB dataset. First row: detections obtained by the proposed method. Second row: detections obtained by using traditional HOG features. Correct detections are indicated by green boxes, while missing ones by dashed yellow boxes and false ones by red boxes. (Color figure online)

3.2 Experiments on the PASCAL VOC 2007 Dataset

The PASCAL VOC 2007 dataset contains a number of visual object classes in realistic scenes. We carried out experiments on four object classes from this dataset, including person, car, bicycle, and dog, since they contain more positive training samples than the others to allow learning without external datasets. We simply trained a general detector for each object class. The trained detectors were evaluated on the test set of PASCAL VOC 2007. Table 1 shows the results of detection by the proposed method in comparison to the results of state-of-the-art algorithms [8, 14, 25–27]. The proposed method obtained the highest detection rates on 2 out of 4 classes. On the bicycle class, the proposed method performs better than other methods except for [14]. On the dog class, our method does not work so well. The reason is that oriented gradient features of furry animals are not so evident and it is easy to mistaken dogs as other furry animals. But our method still has a higher detection rate than the traditional HOG features, which indicates that the learned oriented gradient features can better capture the appearance and shape of furry animals. As for RCNN [26], it applies high-capacity convolutional neural networks to bottom-up region proposals and thus can learn more powerful features to describe objects. However, the detection of RCNN is slow. At test time, features are extracted from each object proposal in each test image. Detection with VGG16 takes 47 s/image on a GPU. Our method only takes 0.4 s/image on a CPU. On the whole, the improvement of our method over HOG features indicates that the learned oriented gradients are better adapted to challenging objects.

3.3 Experiments on the PASCAL VOC 2005 Dataset

In this section, we carried out experiments on side-view car detection based on the PASCAL VOC 2005 dataset. 600 side-view car samples were collected from the UIUC subset and ETHZ subset in PASCAL VOC 2005. We further mirrored the 600 samples to generate 1200 positive training samples. All car samples were resized to 80 × 30 pixels. The proposed method took 39 min to finish training and produced 6 stages in cascade. The test set contains 200 side-view cars in 170 images from the TUGRAZ subset in PASCAL VOC 2005. The detection results are summarized in Table 2, which demonstrate that the proposed method has higher detection rate compared to HOG, SURF, and Haar features.

Table 1. Detection rates on four object classes from the PASCAL VOC 2007 dataset.

Method	Person	Car	Bicycle	Dog
Ours	**48.4**	**60.3**	59.2	11.8
Traditional HOG features	43.2	54.1	55.7	10.9
Viola and Jones [8]	40.4	52.3	52.7	8.1
RCNN fc$_7$ [26]	43.3	58.9	57.9	**46.0**
iCCCP [25]	36.6	51.3	55.8	12.5
HOG-LBP [14]	44.6	58.2	**59.8**	15.1
MILinear [27]	21.9	45.0	39.7	21.3

Table 2. Detection results on side-view cars from the PASCAL VOC 2005 dataset.

	Detection rate	False positives
Ours	**80.5**	21
SURF cascade [24]	70	18
Traditional HOG features	74	29
Viola and Jones [8]	68	34

4 Conclusion

This paper presents a method of learning features from oriented gradients in local patches for robust object detection. Gradients in a local patch are grouped according to their orientations, which leads to 9 patch-maps. Rectangular cells are constructed in each patch-map, which are used to generate candidate local features by summing up magnitudes of oriented gradients therein. The local features are then selected by boosting. Eventually, each local patch is represented by a 90-dimensional feature vector. A higher-level boosting is then carried out over local patches in the detection window. Experimental results on three public datasets demonstrated the superiority of the proposed approach in comparison to existing methods in terms of both training efficiency and detection accuracy.

Acknowledgements. This work was supported by National Natural Science Foundation of China (Nos. 61172141, U1611461, 61573387), Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase), Project on the Integration of Industry, Education and Research of Guangdong Province (No. 2013B090500013), Major Projects for the Innovation of Industry and Research of Guangzhou (No. 2014Y2-00213), and Science and Technology Program of Guangzhou (No. 2014J4100092).

References

1. Coates, A., Ng. A.Y.: Multi-camera object detection for robotics. In: ICRA, pp. 412–419 (2010)
2. Satzoda, R.K., Trivedi, M.M.: Overtaking and receding vehicle detection for driver assistance and naturalistic driving studies. In: ITSC, pp. 697–702 (2014)
3. Fang, S., Tong, S., Xu, X., Xie, Z.: A method of target detection and tracking in video surveillance. In: ICIG, pp. 84–87 (2004)
4. Li, J., Wang, T., Zhang, Y.: Face detection using SURF cascade. In: ICCV, pp. 2183–2190 (2011)
5. Li, L.J., Li, R., Fei-Fei, L.: Towards total scene understanding: classification, annotation and segmentation in an automatic framework. In: CVPR, pp. 2036–2043 (2009)
6. Hayashi, T., Uchiyama, H., Pilet, J., Saito, H.: An augmented reality setup with an omnidirectional camera based on multiple object detection. In: ICPR, pp. 3171–3174 (2010)
7. He, Z., Sun, Z., Tan, T., Qiu, X., Qiu, C., Dong, W.: Boosting ordinal features for accurate and fast iris recognition. In: CVPR, pp. 1–8 (2008)
8. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: CVPR, pp. 511–518 (2001)
9. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3021, pp. 69–82. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24670-1_6
10. Zhu, Q., Yeh, M., Cheng, K., Cheng, S.: Fast human detection using a cascade of histograms of oriented gradients. In: CVPR, pp. 1491–1498 (2006)
11. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV, pp. 1515–1522 (2015)
12. Vedaldi, A., Gulshan, V., Gulshan, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV, pp. 606–613 (2009)
13. Deselaers, T., Ferrari, V.: Global and efficient self-similarity for object classification and detection. In: CVPR, pp. 1633–1640 (2010)
14. Zhang, J., Zhang, K., Yu, Y., Tan, T.: Boosted local structured HOG-LBP for object localization. In: CVPR, pp. 1393–1400 (2011)
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
16. Liao, S., Jain, A.K., Li, S.Z.: A fast and accurate unconstrained face detector. IEEE Trans. PAMI **38**(2), 211–223 (2015)
17. Viola, P.A., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)

18. Jones, V., Miller, E.L.: Online domain adaptation of a pre-trained cascade of classifiers. In: CVPR, pp. 577–584 (2011)
19. Girshick, R., Iandola, F., Iandola, T., Malik, J.: Deformable part models are convolutional neural networks. In: CVPR, pp. 437–446 (2015)
20. Girshick, R.: Fast R-CNN. In: ICCV, pp. 1440–1448 (2015)
21. Yang, F., Choi, W., Lin, Y.: Exploit all the layers: fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In: CVPR, pp. 2129–2137 (2016)
22. Fan, R., Chang, K., Hsieh, C.J., Wang, X., Lin, C.: Liblinear: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
23. Schnitzspan, P., Fritz, M., Roth, S., Schiele, B.: Discriminative structure learning of hierarchical representations for object detection. In: CVPR, pp. 2238–2245 (2009)
24. Li, J., Zhang, Y.: Learning SURF cascade for fast and accurate object detection. In: CVPR, pp. 3468–3475 (2013)
25. Zhu, L., Chen, Y., Yuille, A., Freeman, W.: Latent hierarchical structural learning for object detection. In: CVPR, pp. 1062–1069 (2010)
26. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, pp. 580–587 (2014)
27. Ren, W., Huang, K., Tao, D., Tan, T.: Weakly supervised large scale object localization with multiple instance learning and bag splitting. IEEE Trans. PAMI **38**(2), 405–416 (2016)