A Cascade of Feed-Forward Classifiers for Fast Pedestrian Detection

Yu-Ting Chen[1,2] and Chu-Song Chen[1,3]

[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan
[2] Dept. of Computer Science and Information Engineering,
National Taiwan University
[3] Graduate Institute of Networking and Multimedia, National Taiwan University
{yuhtyng,song}@iis.sinica.edu.tw

Abstract. We develop a method that can detect humans in a single image based on a new cascaded structure. In our approach, both the rectangle features and 1-D edge-orientation features are employed in the feature pool for weak-learner selection, which can be computed via the integral-image and the integral-histogram techniques, respectively. To make the weak learner more discriminative, Real AdaBoost is used for feature selection and learning the stage classifiers from the training images. Instead of the standard boosted cascade, a novel cascaded structure that exploits both the stage-wise classification information and the inter-stage cross-reference information is proposed. Experimental results show that our approach can detect people with both efficiency and accuracy.

1 Introduction

Detecting pedestrians in an image has received considerable attentions in recent years. It has a wide variety of applications, such as video surveillance, smart rooms, content-based image retrievals, and driver-assistance systems. Detecting people in a cluttered background is still a challenging problem, since different postures and illumination conditions can cause a large variation of appearances.

In object detection, both efficiency and accuracy are important issues. In [1], Viola and Jones proposed a fast face detection framework through a boosted cascade. This cascade structure has been further applied to many other object detection problems. For instance, Viola et al. [2] used the cascade framework for pedestrian detection. Rectangle features, which can be evaluated efficiently via the technique of integral image, are employed as the basic elements to construct the weak learners of the AdaBoost classifier for each stage in the cascade. While the use of rectangle features is effective for object-detection tasks such as face detection, they still encounter difficulties in detecting people. It is because that the rectangle features are built by only using intensity information, which is not sufficient to encode the variance of human appearances caused by some factors that can result in large gray-value changes, such as the clothes they wear.

Recently, Dalal and Triggs [3] presented a people detection method with promising detection performances. This method can detect people in a single

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 905–914, 2007.

image. In this work, edge-based features, HOG (*Histograms of Oriented Gradients*), are designed for capturing edge-orientation structure that can characterize human images effectively. HOG features are variant from Lowe's SIFT [4] (*Scale Invariant Feature Transform*), but they are computed on a dense grid of uniform space. Nevertheless, a limitation of this method is that a very high-dimensional feature vector is used to describe each block in an image, which requires a long computation time. To speed up the detection, Zhu et al. [5] combined the above two methods by using linear SVM classifier with HOG features as a weak learner in the AdaBoost stages of the cascaded structure, and enhance the efficiency of the HOG approach.

In this paper, we develop an object detection framework with both efficiency and accuracy. Our approach employs rectangle features and 1-D edge-orientation features that can be computed efficiently. To make the weak learner more discriminative, we use Real AdaBoost as a stage classifier in the cascade. Instead of learning a standard boosted cascade [1] for detection, a new cascading structure is introduced in this paper to exploit not only the stage-wise classification information, but also the inter-stage cross-reference information, so that the detection accuracy and efficiency can be further increased.

2 Previous Work

There are two main types of approaches on pedestrian detection, the holistic approach and the component-based approach. In holistic approaches, a full-body detector is used to analyze a single detection window. The method of Gavrila and Philomin [6] detects pedestrians in images by extracting edge images and matching them to a template hierarchy of learned examplars using chamfer distances. Papageorgiou and Poggio [7] adopted polynomial SVM to learn a pedestrian detector, where Haar wavelets are used as feature descriptors. In [1], Viola and Jones proposed a boosted cascade of Haar-like wavelet features for face detection. Subsequently, this work was further extended to integrating intensity and motion information for walking person detection [2]. Dalal and Triggs [3] designed HOG appearance descriptors, which are fed into a linear SVM for human detection. Zhu et al. [5] employed the HOG descriptor in the boosted cascade structure to speed up the people detector. Dalal et al. [8] further extended the approach in [3] by combining the HOG descriptors with oriented histograms of optical flow to handle space-time information for moving human.

The holistic approaches may fail to detect pedestrians when occlusion happens. Some component-based researches were proposed to deal with the occlusion problem. Generally, a component-based approach searches for a pedestrian by looking for its apparent components instead of the full body. For example, Mohan et al. [9] divided the human body into four components, head, legs, and left/right arms, and a detector is learned by using SVM with Haar features for each component. In [10], Mikolajczyk et al. used position-orientation histograms of binary edges as features to build component-based detectors of frontal/profile heads, faces, upper bodies, and legs. Though component-based approaches can cope with the occlusion problem, a high image resolution of the detection window is required for capturing sufficient information of human components. This restricts the range of applications. For example, the resolution of humans in some surveillance videos is too low to detect by component-based approaches.

In this paper, we propose a holistic human detection framework. Our approach can detect humans in a single image. It is thus applicable for the cases where only single images are available, such as detecting people in home photos.

The rest of this paper is organized as follows: In Section 3, the Real AdaBoost algorithm using rectangle features and EOH fetures is introduced. A novel cascaded structure of feed-forward classifiers is proposed in Section 4. Experimental results are shown in Section 5. Finally, a conclusion is given in Section 6.

3 Real AdaBoost with Rectangle and EOH Features

3.1 Feature Pool

Features based on edge orientations have been shown effective for human detection [3]. In the HOG feature [3], each image block is represented as 7 × 15 overlapping sub-blocks. Each sub-block contains 4 non-overlapping regions, where each region is represented as a 9-bin histogram with each bin being corresponding to a particular edge orientation. In this way, a 3780-dimensional feature, encoding part-based edge-orientation distribution information, is used to represent an image block. Such a representation is powerful for people detection, but has some limitations. First, the representation is too complex to evaluate, and thus the detection speed is slow. Second, all of the dimensions in an HOG feature vector are employed simultaneously, declining the chance of employing only part of them, which may be capable of rejecting the non-human blocks, for fast pre-filtering. A high-dimensional edge-orientation feature like HOG can be treated as a combination of many low-dimensional ones. In our approach, instead of employing a high-dimensional feature vector, we use a set of one-dimensional features derived from edge orientations, as suggested by Levi and Weiss [11].

Similar to HOG, the EOH (*Edge Orientation Histogram*) feature introduced in [11] also employs the edge-orientation information for feature extraction, but an EOH feature can characterize only one orientation at a time, and each EOH feature is represented by a real value. Unlike the HOG that is uniquely defined for an image region, many EOH features (with respect to different orientations) can be extracted from an image region, and each of which is only of one-dimension. Therefore, there is a pool of EOH features allowed to be selected for a region. The EOH feature is thus suitable to be integrated into the AdaBoost or boosted-cascade approaches for weak-learner selection.

In our approach, the EOH feature is employed in the AdaBoost stage of our cascading structure. Since the weak learners employed are all one-dimension with the output being simply scalars, the resulted AdaBoost classifier is more efficient to compute than which of using high-dimensional features (e.g. HOG) for building the weak learners [5]. We briefly review the EOH features in the following.

To compute EOH features, the pixel gradient magnitude m and gradient orientation θ in a block B are calculated by *Sobel* edge operator. The edge orientation is evenly divided into K bins over 0° to 180° and the sign of the orientation is ignored, and thus the orientations between 180° to 360° are considered as the same to those between 0° to 180°. Then, the edge orientation histograms $E_k(B)$ in each orientation bin k of block B is built by summing up all of the edge magnitudes whose edge orientations are belonging to bin k. The EOH features we adopted is measured by the ratio of the bin value of a single orientation to the sum of all the bin values as follows:

$$F_k(B) = \frac{E_k(B) + \epsilon}{\sum_i E_i(B) + \epsilon}, \tag{1}$$

where ϵ is a small positive value to avoid the denominator being zero. Each block thus has K EOH features, $F_1(B), \ldots, F_K(B)$, which are allowed to be selected as weak learners. Similar to the usage of the integral image technique for fast evaluating the rectangle features, integral histogram [12] can be used to efficiently compute the EOH features.

The feature pool employed in our approach for AdaBoost learning contains the EOH features. To further enhance the detection performance, we also include the rectangle features used in [1] for weak learner selection.

3.2 Learning Via Real AdaBoost

After forming the feature pool, we learn an AdaBoost classifier for some stages of our cascading structure. Typically, the AdaBoost alrogithm selects weak learners of binary-valued outputs obtained by thresholding the feature values as shown in Fig. 1(a) [1,2,5,11]. However, a disadvantage of the thresholded-type weak learners is that it is too crude to discriminate the complex distributions of the positive and negative training data. To deal with the problem, Schapire et al. [13] suggested the use of the Real AdaBoost algorithm. To represent the distributions of positive and negative data, the domain space of the feature value is evenly partitioned into N disjoint bins (see Fig. 1(b)). The real-valued output in each bin is calculated according to the ratio of the training data falling into the bin. The weak learner output then depends only on the bin to which the input data belongs. Real AdaBoost has shown its better discriminating power between positive and negative data [13]. This algorithm is employed to find an AdaBoost classifier for each stage of the cascade, and more details can be found in [13].

4 Feed-Forward Cascade Architecture

The Viola and Jones cascade structure containing S stages is illustrated in Fig. 2, where $\mathbf{A}_i$ is referred to as an AdaBoost or Real AdaBoost classifier in the i-th stage. In this cascaded structure, negative image blocks that do not contain humans can be discarded in some early stages of the cascade. Only the blocks passing all the stages are deemed as positive ones (i.e., the ones containing

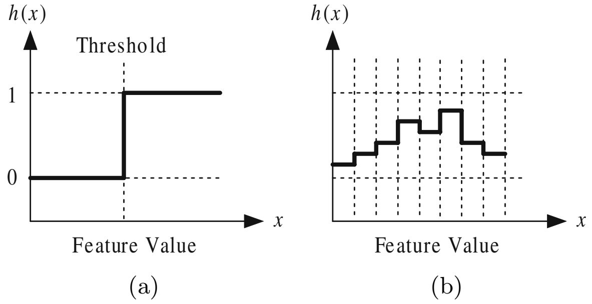


Fig. 1. Two types of weak classifiers: (a) Binary-valued weak classifier and (b) Real-valued weak classifier

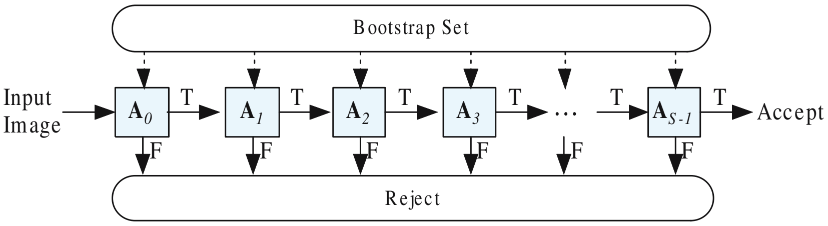


Fig. 2. Viola and Jones cascade structure. To learn each stage, negative images are randomly selected from the bootstrap set as shown in dashed arrows.

humans). A characteristic of the cascading approach is that the decision time of negative and positive blocks are un-equal, where the former takes less but the later takes much. To find an object of unknown positions and sizes in an image, it usually involves the search of the blocks of all possible sites and scales in the image. In this case, since the negative blocks required to be verified in an image are usually far more than the positive blocks, saving the detection time of the negative blocks thus increases the overall efficiency of the object detector.

To train such a cascaded structure, we usually set a goal for each stage. The later the stages, the more difficult the goals. For example, consider the situation that the first stage is designed with 99.9% positive examples being accepted and 50% negative examples being rejected. Then, in the second stage, the positive examples remain the same, but the negative examples include those not successfully rejected in the bootstrap set by the first stage. If we set the goal of the second stage again as accepting 99.9% positive examples and rejecting 50% negative examples, respectively, and repeat the procedure for the later stages, the accepting rate of positive examples and rejecting rate of negative examples in i-th stages are $(99.9\%)^i$ and $1 - (50\%)^i$ respectively for the training data. In each stage, the Real AdaBoost algorithm introduced in Section 3.2 can be used to select a set of weak learners from the feature pool to achieve the goal. Since more difficult negative examples are sent to the later stages, it usually happens that more weak learners have to be chosen to fulfill the goal in the later stage.

The degree of accurate prediction in each stage is evaluated by the confidence score. A high confidence value implies an accurate prediction. Each stage learns its own threshold to accept or reject an image block as shown in Fig. 3(a). In Viola and Jones structure, the confidence value is discarded in successive stages. That is, once the confidence value is employed to make a binary decision (yes or no) in the current stage, it will be no longer used in the later stages. This means that the stages are independent to each other and no cross-stage references are allowed. Nevertheless, exploiting the inter-stage information is possible to boost the classification performance further. It is because that, by compositing the confidence values of multiple stages (say, d-stages) as a vector and making a decision in the d-dimensional space, the classification boundaries being considered will not be restricted as hyper-planes parallel to the axes of the stages (as shown in Fig. 3(a)), but can be hyper-planes (or surfaces) of general forms. A two-dimensional case is illustrated in Fig. 3(b).

One possible way to exploit the inter-stage information is to delay the decision making of all the S stages in the cascade, and perform a post-classification in the S-dimensional space to make a unique final decision. However, making a decision after gathering all the confidence scores will considerably decrease the detection efficiency, since there is no chance to early jump out the cascade. In this paper, we propose a novel approach that can exploit the inter-stage information while preserve the early jump-out effect.

4.1 Adding Meta-stages

Our method is based on adding some *meta-stages* in the original boosted cascade as shown in Fig. 4. A meta-stage is a classifier that uses the inter-stage information (represented as the confidence scores) of some of the previous stages for learning. Like an AdaBoost stage, a meta-stage is also designed with a goal to accept and reject the pre-defined ratios of positive and negative examples respectively, and the prediction accuracy is also measured by the confidence score of the adopted classification method for the meta-stage.

In our approach, the meta-stages and the AdaBoost stages aligned in the cascade are designed as **AAMAMAM**... **AM**, where '**A**' and '**M**' denote the AdaBoost stages and meta-stages, respectively, as shown in Fig. 4. In this case, the meta-stage is a classifier in the two-dimensional space. The input vector of the first meta-stage $\mathbf{M}_1$ is a two-dimensional vector $(C(\mathbf{A}_0), C(\mathbf{A}_1))$, where $C(\mathbf{A}_i)$ is the confidence score of the i-th AdaBoost stage. The input vector of the other meta-stage $\mathbf{M}_i (i = 2, \ldots, H)$ is also a two-dimensional vector $(C(\mathbf{M}_{i-1}), C(\mathbf{A}_i))$ that consists of the confidence values of the two closest previous-stages in the cascade, where $C(\mathbf{M}_i)$ is the confidence score of the i-th meta-stage.

The meta-stage introduced above is light-weight in computation since only a two-dimensional classification is performed. However, it can help us to further reject the negative examples during training the entire cascade. In our implementation, we usually set the goal of the meta-stage as allowing all the positive training examples to be correctly classified, and finding the classifier with the highest rejection rate of the negative training exmaples under this condition.

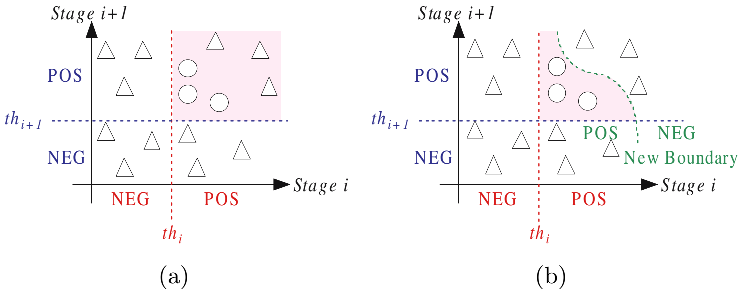


Fig. 3. Triangles and circles are negative and positive examples shown in data space. (a) The data space is separated into object (POS) and non-object (NEG) regions by thresholds th_i and th_{i+1} in stages i and $i+1$. (b) The inter-stage information of stage i and $i+1$ can be used to learn a new classification boundary as shown in green line.

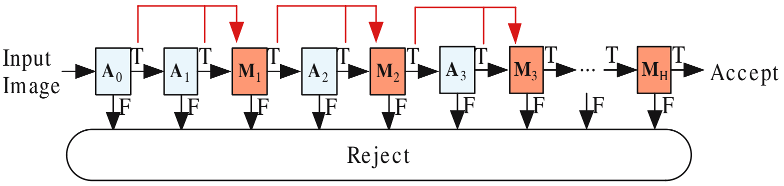


Fig. 4. Feed-forward cascade structure

This criterion will not influence the latest decision of the previous AdaBoost classifier about the positive data, but can help reject more of the negative data. In our experience, by adding the meta stages, the total number of the required AdaBoost stages can be reduced when the same goals are set to be fulfilled.

4.2 Meta-stage Classifier

The classification method used in the meta-stage can be arbitrary. In our work, we choose the linear SVM as the meta-stage classifier due to its high generalization ability and efficiency in evaluation. To train the meta-stage classifier, 3-fold cross-validation is applied for selecting the best penalty parameter C of the linear SVM. Then, a maximum-margin hyperplane that separates the positive and negative training data can be learned. To achieve the goal of the meta-stage, we move the hyperplane along its normal direction by applying different thresholds, and find the one with the highest rejection rate for the negative training data (under the situation that no positive ones will be falsely rejected). Note that, even a two-dimensional classifier is used, each meta-stage inherently contains the confidence of all the previous stages. This is because that a meta-stage (except to the first one) employs the confidence value of its closest previous meta-stage as one of the inputs. Thus, information of the previous stages will be iteratively feed-forwarded to the later meta-stages.

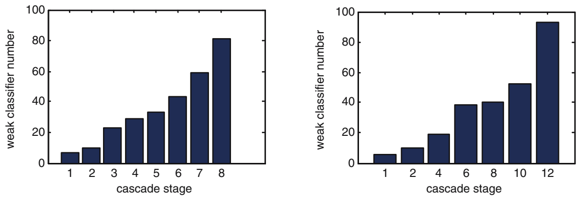
5 Experimental Result

To evaluate the proposed cascade structure, a challenging pedestrian data set, INRIA person data set [3], is adopted in our experiments. This data set contains standing people with different orientations and poses, in front of varied cluttered backgrounds. The resolution of human images is 64×128 pixels. Within a 64×128 detection block, the feature pool contains 22477 features (6916 rectangle features and 15561 EOH features) for learning the AdaBoost stages, and the domain space of the feature value is evenly divided into 10 disjoint bins for each feature in the Real AdaBoost algorithm. The edge orientation is evenly divided into 9 bins over 0° to 180° to calculate the EOH feature. A bootstrap set with 3373860 negative images is generated by selecting sub-images from the non-pedestrian training images in different positions and scales. We refer the method presented in Section 3 the ErR-cascade method since it employs the EOH and rectangle features in the Real AdaBoost algorithm for human detection. The method where the meta-stages are further added (as illustrated in Fig. 4) is referred to as the ErRMeta-cascade method. In the ErRMeta-cascade method, all meta-stages are two-dimensional classifiers, and the linear SVM is adopted as the meta-stage learners. Thus the meta-stages can be computed very fast since only a two-dimensional inner product is needed.

We use the same number of positive and negative examples for training each stage of the cascade: The data set provides 2416 positive training data and we randomly select 2416 negative images from the bootstrap set as the negative training data. In training each AdaBoost stage, we keep adding weak learners until the predefined goals are achieved. In our experiments, we require that at least 99.95% positive examples are accepted and at least 50% negative examples are rejected in each AdaBoost stage. For meta-stages, we only require that all the positive examples should be accepted and find the classifier with the highest negative-example rejection rate. If the false positive rate of the cascade is below 0.5%, the cascade structure will stop learning new stages.

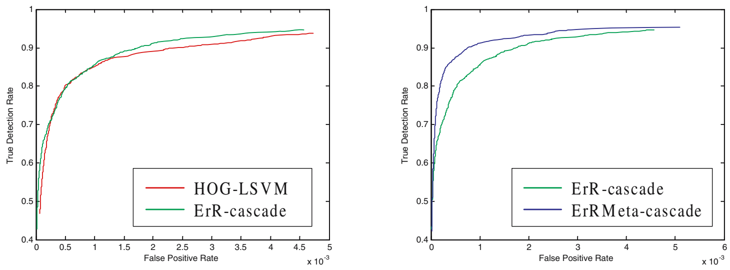
We also implemented the Dalal and Triggs method [3] (referred to as the HOG-LSVM method). First, we compare the performances of the ErR-cascade and the HOG -LSVM method. After training, there are eight AdaBoost stages with 285 weak classifiers in the ErR-cascade as shown in Fig. 5(a). For a 320×240 image (containing 2770 detection blocks), the averaged processing speeds of the HOG-LSVM and the ErR-cascade are 1.21 and 9.48 fps (frames per second) respectively by using a PC with a 3.4 GHz processor and 2.5 GB memory. Since the HOG-LSVM uses a 3780-dimensional feature, their method is time-consuming. As to the detection result, the ROC curves of these two methods are shown in Fig. 6(a). From the ROC curves, the detection result of the ErR-cascade is in overall better than that of the HOG-LSVM method. The introduced ErR-cascade method thus highly improves the detection speed and also slightly increases the detection accuracy than the HOG-LSVM method.

Then, we compare these methods to the method with meta-stages. All the goal settings of the AdaBoost stages are the same as those of ErR-cascade. After training, there are seven AdaBoost stages with 258 weak classifiers as



(a) (b)

Fig. 5. The number of weak classifiers learned in each AdaBoost stage of the ErR-cascade method (a) and ErRMeta-cascade method (b)



(a) (b)

Fig. 6. (a) The ROC curves of the HOG-LSVM method and the ErR-cascade method. (b) The ROC curves of the ErR-cascade method and ErRMeta-cascade method.

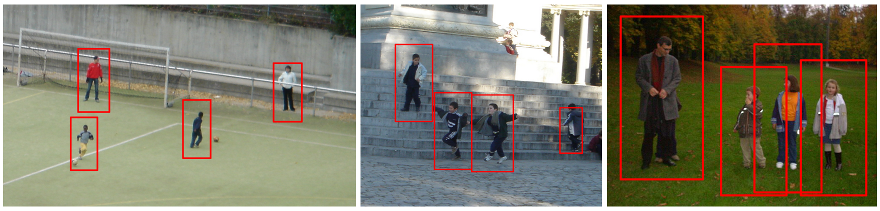


Fig. 7. Experimental results of the ErRMeta-cascade method

shown in Fig. 5(b) and six meta-stages. For a 320 × 240 image, the averaged processing speed is 10.13 fps. For the ErR-cascade method, the trained cascade contains less weak learners, and some non-pedestrian blocks can be early rejected by the cascade in the meta-stages with less computation. The ROC curves of ErR-cascade and ErRMeta-cascade are shown in Fig. 6(b). The results demonstrate that, by adding the meta-stages, both the detection speeds and accuracies can be further raised. Some results are shown in Fig. 7.

6 Conclusion

A novel cascaded structure for pedestrian detection is presented in this paper, which consists of the AdaBoost stages and meta-stages. In our approach, the 1-D EOH edge-based feature is employed for weak-learner selection and Real AdaBoost algorithm is used as the AdaBoost-stage classifier to make the weak learner more discriminative. As to the meta-stages, the inter-stage information of the previous stages is composed as a vector for learning a SVM hyperplane, so that the negative examples can be further rejected. Based on experimental results, our approach is practically useful since it can detect pedestrian with both efficiency and accuracy.

Although the cascade type of **AAMAMAM**... **AM** is used, our approach can be generalized to other types to composite the AdaBoost stages and the meta-stages. In the future, we plan to employ our method for other object detection problems, such as faces, vehicles, and motorcycles.

Acknowledgments. This work was supported in part under Grants NSC96-2422-H-001-001.

References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE CVPR, vol. 1, pp. 511–518 (2001)
2. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: IEEE ICCV, vol. 2, pp. 734–741 (2003)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE CVPR, vol. 1, pp. 886–893 (2005)
4. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
5. Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S.: Fast human detection using a cascade of histograms of oriented gradients. In: IEEE CVPR, vol. 2, pp. 1491–1498 (2006)
6. Gavrila, D., Philomin, V.: Real-time object detection for "smart" vehicles. In: IEEE ICCV, vol. 1, pp. 87–93 (1999)
7. Papageorgiou, C., Poggio, T.: A trainable system for object detection. IJCV 38(1), 15–33 (2000)
8. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, Springer, Heidelberg (2006)
9. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. IEEE PAMI 23(4), 349–361 (2001)
10. 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, pp. 69–82. Springer, Heidelberg (2004)
11. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: the importance of good features. In: IEEE CVPR, vol. 2, pp. 53–60 (2004)
12. Porikli, F.: Integral histogram: a fast way to extract histograms in cartesian spaces. In: IEEE CVPR, vol. 1, pp. 829–836 (2005)
13. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning 37(3), 297–336 (1999)