Building Feature Pool Effectively for Real-Time Pedestrian Detection

Jijun Yang, Yingdong Ma and Zhibin Zhang

College of Computer Science, Inner Mongolia University, No.235 Daxuexilu, Hohhot, China
Email: 284690299@qq.com, csmyd@imu.edu.cn, cszhbin@imu.edu.cn

Abstract. When using AdaBoost algorithm to train a classifier for visual object detection, it is important to build a proper feature pool. Usually, there are a large number of features available for this work. Current top performance detectors either utilize all features or select part of them due to the limited computer memory. In this paper, we discuss how to build feature pool effectively for real-time object detection. In particular, we focus on the problem of pedestrian detection using integral channels features. We demonstrate that a small feature pool is sufficient to build a high quality classifier. The main advantage of this work is that the training time can be reduced drastically. According to the experimental results, for a test image with $640 \times 480$ pixels, the proposed detector runs at over 30 fps with competitive quality which makes it suitable for real-time applications.

1. Introduction

Fast and high quality object detection enables a broad range of applications, which has motivated a large amount of research on the topic. In this paper we focus on pedestrian detection because it plays an important role in various applications, such as car safety, surveillance, and robotics, etc. To achieve better performance, most detection methods either applies multiple hand-crafted features [1] or convolutional features [2]. Our work is based on the integral channel features, which is introduced by Dollár et al. in [3]. We use Ada Boost Learning algorithm to build the baseline pedestrian detection classifier. Although deep neural network based methods are more popular recently [4,5], this detection framework is still being studied and achieves surprisingly competitive object detection results [6-9].

Integral Channel Features detector can be seen as a combination of the classic Viola and Jones work (VJ) [10] with Dalal and Triggs' work (HOG+SVM) [11]. Given an input image, a set of gradient and color “channels” are computed. Then low level features are built by summing over rectangular regions.

A lot of research works have been invested on Integral Channel Features. Most of these works focus on additional data [12], better classifier [13], deformable parts [14], multi-scale models [15], and combination with deep neural networks [8]. Despite these valuable works, integral channel features still leave some open questions. For example, how many features should we use in a pedestrian detection system? How to select features for more efficient training? To our best knowledge, there is no research work analyzes feature selection for efficient classifier training. For a $64 \times 128$ pixels model (with shrinking factor 4), there are $718080$ rectangle features in total [15]. It is a time-consuming task for most desktop computers to use all rectangle features for training. Therefore, in this work, we try to find answers for the questions: How to select features for real-time pedestrian detection? Does a classifier with more features must be better than other methods with less features?

In the proposed method, computing of integral channels, feature responses, and detection scores are
parallel implemented for improving system operation efficiency. On a single CPU+GPU desktop computer, our pedestrian detector runs at over 30 fps for images with 640×480 pixels. The method is suitable for real-time applications, such as the automatic pilot system for a driverless vehicle.

1.1. Contributions

- We perform a detailed investigation on feature sizes and discuss the relationship between feature size and detection effectiveness.
- We demonstrate that, by selecting features properly, a small feature pool with less features is sufficient to build a high performance, real-time classifier.

Sections 2 describes our base detector and the experimental setup. Section 3 and section 4 explore feature size and feature numbers respectively. These key factors should be considered carefully when building the feature pool. Finally, we conclude this work in section 5.

2. Baseline Detector and Experimental Setup

2.1. Baseline Detector

Dollár et al. introduced the integral channel features (ICF) [3], in which HOG and LUV channels are computed so that ICF consists of 10 channels in total. The features are assembled into a linear combination of 2048 level-2 decision trees (containing 3 stumps) using the Adaboost algorithm. Training is done in 4 stages, which starts from a classifier with 512 trees and followed by 1024, 2048, and 2048 trees respectively. The first stage randomly samples 5000 negative samples. In each of the next three stages, 5000 additional hard negatives are added to the training system.

The learned strong-classifier is applied at test time as a sliding window over test images. The model size is 64×128 pixels. To make the algorithm faster with no significant impact on detection quality, coordinates of the feature pool and the candidate detection windows can be quantized by a factor of 4 (so called “shrinking factor”). To obtain the final detections, a greedy non-maximal suppression method is used [16].

2.2. Experimental Setup

For evaluation we use the Caltech pedestrian detection benchmark. This benchmark provides software for evaluation and comparison state-of-the-art methods over multiple datasets. This software takes care of several issues when evaluating detection algorithms [17].

All our models are trained and tested using the Caltech-USA pedestrian dataset. There are 1631 pedestrians over 4250 frames in training, 1014 pedestrians over 4024 frames in testing. Evaluation of the experiments is done in terms of false positives per image (FPPI, not per window) [17]. In the following sections, we will report miss rate at the reference point of 10-1 FPPI.

3. Feature Size

We know that a large number of features increase computational burden and lead to an inefficient detection system. Thus, on the premise of maintaining system performance, what is the optimal feature size? In order to find the answer of this question, we study rectangular features of all sizes within 8×8 pixels. We trained 64 classifiers, each of them uses a single size feature (see table 1). Because of the shrinking factor, in this method every 4 pixels means size 1.
Table 1. Miss rate of 64 classifiers

| Feature Width | Feature Height | H=1   | H=2   | H=3   | H=4   | H=5   | H=6   | H=7   | H=8   |
|---------------|----------------|-------|-------|-------|-------|-------|-------|-------|-------|
| W=1           |                | 34.22%| 27.15%| 27.69%| 26.05%| 28.28%| 29.20%| 33.29%| 32.81%|
| W=2           |                | 28.92%| 26.60%| 23.18%| 23.51%| 26.17%| 30.32%| 31.03%| 34.73%|
| W=3           |                | 28.76%| 24.96%| 23.82%| 25.96%| 28.77%| 29.90%| 32.27%| 36.60%|
| W=4           |                | 29.80%| 25.93%| 26.66%| 28.97%| 30.20%| 34.13%| 34.85%| 39.11%|
| W=5           |                | 31.69%| 30.71%| 32.07%| 29.23%| 33.66%| 36.01%| 37.43%| 41.86%|
| W=6           |                | 35.52%| 34.96%| 35.65%| 36.07%| 38.53%| 41.28%| 41.76%| 45.88%|
| W=7           |                | 41.68%| 39.14%| 40.79%| 38.82%| 43.94%| 44.23%| 47.81%| 48.96%|
| W=8           |                | 45.52%| 45.54%| 42.61%| 44.37%| 46.08%| 49.34%| 51.46%| 52.39%|

Table 1 shows that the best feature size is 2×3 pixels, at which Miss Rate (MR) is 23.18%. We find that if two features have the same feature area, the narrow feature is usually better than the wide feature. For example, the feature with size of 3×4 pixels is better than the feature with size of 4×3 pixels. In addition, as the feature size increases, the detection result becomes better until 3×3 pixels. Since then, larger feature size leads to worse performance (Figure 1).

![Figure 1](image)

Figure 1. Examples of different feature sizes and their miss rate

4. Number of Features
The main bottleneck of current top quality pedestrian detection methods is the problem of using too many features, which generates heavy memory load during training time. In the case of large image size, it might reach computer hardware limitation. The number of rectangles inside a 64 ×128 pixels model is very large even using a shrinking factor of 4. Then how many features should we use during training? In general, a classifier with more features has better performance than other methods with fewer features. However, we find that if select features carefully, a classifier with fewer features can achieve high performance as well.

In the next experiments, we use a desktop computer with Intel Core i5-7640X CPU(4.0GHz), 32 GB memory and a GeForce GTX 1060 3GB GPU. The following features are adopted in these experiments:
- AllSquares: All square features in the range of 16×16 pixels (1×1, 2×2, ..., 16×16). The total number of features is 36720. The training time is about 1 hour and the miss rate is 24.02%.
- AllRectangles: All rectangle features within 8×8 pixels. In this case, the total number of features is 228000. A desktop computer takes about 6 hours to train the classifier which has the miss rate of 24.00%.

In another experiment, rectangle features are limited to the size of 2×3 pixels so that the number of features is 4500. The training time reduced significantly to 20 minutes. To our surprise, even in the case of 4500 features, the classifier achieves 23.18% miss rate. Therefore, our conclusion is that a small feature pool can also obtain a good detection result.

4.1. How to Build the Feature Pool
We take the approach of gradually increasing the number of features to build the feature pool. We start with the best feature size of 2×3 pixels (see Table.1) and then add the second-best feature of size 2×4 pixels. This process continues until we achieve the lowest miss rate.

From Table 2, we can find that there is no obvious connection between the feature pool size and the detection performance. As a result, it is unnecessary to use all rectangle features for training. In our experiment, we start from the best single feature size of 2×3 pixels which has 23.18% miss rate. As more features added, lower detection miss rate can be gained. Finally, using feature pool No.9 with 38970 features, we get the best result of 20.34% miss rate. On the Caltech dataset, it takes 1 hour and 10 minutes for training with 4 rounds.

Figure 2 illustrates performance comparison of our method with state-of-the-art methods. The proposed method achieves 20.34% miss rate, 1.87% higher than the Checkerboards method. Please note that our method achieves the competitive result with a small feature pool (38970 features) which allows it runs at about 30fps on a desktop computer whereas the speed of Checkerboards method is less than 8 fps on the same computer.

| NO. | Feature Pool | Feature Pool Size | Miss Rate  |
|-----|--------------|-------------------|------------|
| 1   | (2,3)        | 4500              | 23.18%     |
| 2   | (2,3),(2,4)  | 8850              | 22.51%     |
| 3   | (2,3),(2,4),(3,3) | 13050            | 22.21%     |
| 4   | (2,3),(2,4),(3,3),(3,2) | 17390            | 21.67%     |
| 5   | (2,3),(2,4),(3,3),(3,2),(4,2) | 21420            | 22.73%     |
| 6   | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4) | 25480            | 21.04%     |
| 7   | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4) | 30120            | 21.62%     |
| 8   | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4),(2,5) | 34320            | 22.73%     |
| 9   | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4),(2,5),(2,2) | 38970            | 20.34%     |
| 10  | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4),(2,5),(2,2),(4,3) | 42870            | 22.78%     |
| 11  | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4),(2,5),(2,2),(4,3),(1,2) | 47830            | 21.45%     |
| 12  | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4),(2,5),(2,2),(4,3),(1,2),(1,3) | 52630            | 23.21%     |
| 13  | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4),(2,5),(2,2),(4,3),(1,2),(1,3),(1,5) | 57110            | 23.41%     |
| 14  | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4),(2,5),(2,2),(4,3),(1,2),(1,3),(1,5),(3,1) | 61590            | 22.33%     |
| 15  | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4),(2,5),(2,2),(4,3),(1,2),(1,3),(1,5),(3,1),(3,5) | 65510            | 21.99%     |
| 16  | (2,3),(2,4),(3,3),(3,2),(4,2),(3,4),(1,4),(2,5),(2,2),(4,3),(1,2),(1,3),(1,5),(3,1),(3,5),(2,1) | 70310            | 23.75%     |
5. Conclusion
A noticeable trend is that researchers increasingly rely on huge feature pools since it is commonly believed that more features integrate more information and thus lead to better performance. Through this paper we have shown that it is unnecessary to use all rectangle features for training because it is not only unhelpful to the improvement of detection quality, but also increase computing load greatly. In this work, we applied the following method to find the optimal size of feature pool. Firstly, we find the best performance single feature within a range of 8×8 pixels. Secondly, the system achieves lower miss rate as more features are added. Finally, the best performance is obtained when the feature pool increased to 38970 features. Experimental results demonstrate that the proposed pedestrian detection method not only reduces training time significantly, but also leads to a better classifier.

6. Acknowledgment
This research was supported by the National Natural Science Foundation of China under Grant 61461039 and Grant 31760345.

7. References
[1] Ma Y, Deng L, Chen X and Guo N 2013 Integrating orientation cue with EOH-OLBP-Based multilevel features for human detection J. IEEE Transactions on Circuits and Systems for Video Technology 23(10) 1755-66
[2] Zhang L, Lin L, Liang X and He K 2016 Is faster R-CNN doing well for pedestrian detection? European Conf. on Computer Vision ed Springer and Cham pp 443 -57
[3] Dollár P, Tu Z, Perona P and Belongie S 2009 Integral channel features Proc. of the British Machine Vision Conf.
[4] Du X, El-Khamy M, Lee J and Davis L 2017 Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection Applications of Computer Vision (2017 IEEE Winter Conference) pp 953-61
[5] Ren S, He K, Girshick R and Sun J 2015 Faster R-CNN: Towards real-time object detection with region proposal networks Advances in Neural Information Processing Systems pp 91-9
[6] Zhang S, Benenson R and Schiele B 2015 Filtered channel features for pedestrian detection Computer Vision and Pattern Recognition (2015 IEEE Conference) pp 1751-60
[7] Kieritz H, Becker S, Hübner W and Arens M 2016 Online multi-person tracking using integral channel features Advanced Video and Signal Based Surveillance (2016 13th IEEE International Conference) pp 122-30
[8] Cao J, Pang Y and Li X 2017 Learning Multilayer Channel Features for Pedestrian Detection J. IEEE Transactions on Image Processing 26 (7) 3210-20
[9] Parate M R, Satpute V R and Bhurchandi K M 2018 Global-patch-hybrid template-based arbitrary object tracking with integral channel features J. Applied Intelligence 48(2) 300-14
[10] Viola P and Jones M J 2004 Robust real-time face detection J. International journal of computer vision 57(2) 137-54.
[11] Dalal N and Triggs B 2005 Histograms of oriented gradients for human detection Computer Vision and Pattern Recognition (IEEE Computer Society Conference) Vol 1 pp 886-93
[12] Zhang S, Benenson R, Omran M, Hosang J H and Schiele B 2016 How far are we from solving pedestrian detection? Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp 1259-67
[13] Benenson R, Omran M, Hosang J H and Schiele B 2014 Ten years of pedestrian detection, what have we learned? European Conference on Computer Vision ed Springer and Cham pp 613-27
[14] Felzenszwalb P, Mcallester D and Ramanan D 2008 A discriminatively trained, multiscale, deformable part model Computer Vision and Pattern Recognition (IEEE Conference) pp 1-8
[15] Benenson R, Mathias M, Tuytelaars T and Van Gool L 2013 Seeking the strongest rigid detector Computer Vision and Pattern Recognition (IEEE Conference) pp 3666-73
[16] Hosang J, Benenson R and Schiele B 2016 A convnet for non-maximum suppression German Conf. on Pattern Recognition ed Springer and Cham pp 192-204
[17] Dollár P, Wojek C, Schiele B and Perona P 2012 Pedestrian detection: An evaluation of the state of the art J. IEEE transactions on pattern analysis and machine intelligence 34(4) 743 - 61