Pedestrian Detection via Combined Cascades

Yujia Tang
Alibaba Group, Hangzhou, China
yujia.tyj@alibaba-inc.com

Abstract

We describe a method of combined cascades architecture for pedestrian detection. In order to keep high performance of the detector and to filter out a vast amount of background faster, we combine hard cascade and soft cascade classifiers. We propose a method, the average judging trees, to compute the shortest length of classifiers. Through the shortest-length soft cascade, the detector generates candidate key positions in an image. Then the detector uses a few parts of hard cascade to obtain target pedestrian positions. The experiments show that our method can balance the trade-off between accuracy and speed for a given classifier.

1 Introduction

The trade-off between accuracy and speed is a central aspect of the problem of object detection in images. Since the object can appear potentially anywhere, and the background region usually cover the most area in the image, the classification function must be applied at a comprehensive set of positions and scales. Furthermore, accurate classification is complex and slow due to the vast variation of appearances of the object, and even greater variation of the non-object class. A good object detector should locate the target accurately and can filter out the background region fast.

A common method of speeding up pedestrian detection is that constructing a cascade classifier with a series of sub-classifiers. For the problem of pedestrian detection, each sub-classifier, or stage, in a cascade, is a binary classification function that is trained to reject a significant fraction of the non-pedestrians, while allowing almost all the pedestrians to pass to the next stage. Each successive stage is trained based on the non-pedestrians that pass all prior stages. The computational speed-up is achieved by weeding out the vast majority of non-pedestrians in the early stages which are relatively simple to evaluate. Thus, each stage in a cascade have a coarse-to-fine ability to filter out background and discriminate pedestrians. The example of pedestrian detection is shown in Figure 1.

The cascaded classifiers is not a new idea. Fast object detection has been of considerable interest in the research community. Notable efforts for increasing detection speed include work by [Felzenszwalb et al., 2013] and Pedersoli et al. [Pedersoli et al., 2011] on cascaded and coarse-to-fine deformable part models, respectively, Lampert et al.'s application of branch and bound search for detection [Lampert et al., 2009], and Dollár et al.'s work on crosstalk cascades [Dollár et al., 2012]. Although recent deep convolutional networks have achieved high performance on object detection, deep learning algorithms are difficult to implement on low power consumption hardware. Deep learning algorithms such as Fast R-CNN [Girshick, 2016], Faster R-CNN [Ren et al., 2015] and Yolo series [Redmon et al., 2016; Redmon and Farhadi, 2017], must use parallel implementation like GPU to achieve fast detection while using rich representations but at the cost of added complexity and hardware requirements.

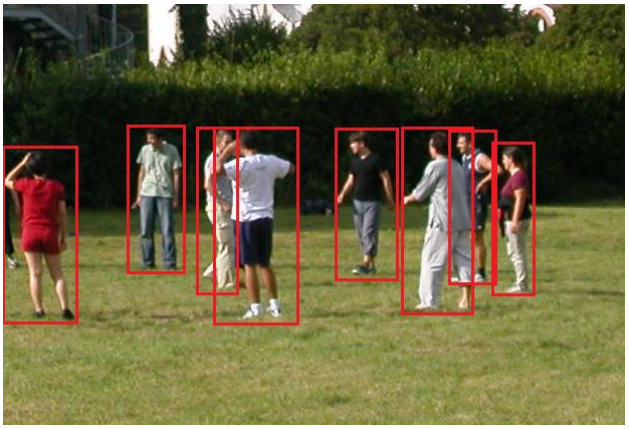

Figure 1: Examples of pedestrians detected by combined cascades classifier.

Common pedestrian detection algorithms with cascade are hard cascade [Viola and Jones, 2005] and soft cascade [Bourdev and Brandt, 2005]. Hard cascade classifier have a higher accuracy on pedestrian regions, while soft cascade can wipe out background regions in an image faster. In this paper, we propose a new algorithm of combining soft cascade classifier and hard cascade classifier with average judging trees to wipe out background regions faster as well as keep high accuracy on pedestrian regions.

The rest of this paper is organized as follows. We describe baseline hard cascade and soft cascade training implementation in section 2 and section 3. In section 4 we describe our

combination method including the new defined algorithm of average judging trees. Finally, in section 5, we compare accuracy and speed to existing approaches. We conclude in section 6.

2 Description of Hard Cascade Classifier

Hard cascade classifier is shown in figure 2. In hard cascade classifier, there are many stages. Some weak classifiers with weights construct every stage. One candidate sample can go into the next stage only when current stage judges it as a positive one. Moreover, in any stage, the candidate sample could be judged as a negative one if the feature value lower than the threshold. Only when the sample passes all the stages, the hard cascade classifier will judge it as the target pedestrian region.

Thus, in a hard cascade classifier, every stage must have a high accuracy on judging the true positive sample, and every stage must have more powerful judging ability than the last one.

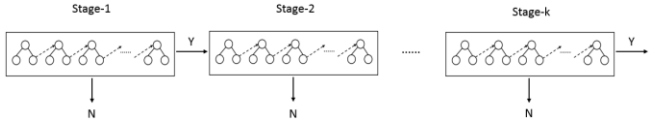

Figure 1: Workflow of hard cascade classifier.

Algorithm 1 Training process of hard cascade detector

Input:

1.Training data set $\{(x_1, y_1), \ldots\ldots, (x_{a+b}, y_{a+b})\}$, a,b is the number of positive and negative samples, respectively.

2.The number of cascades, N.

Initialize:

1.$\omega_{0,y_i} \leftarrow \frac{1}{2a}, \frac{1}{2b}$, initialize the weights of positive and negative samples, in which $y_i = 0,1$.

Train

0: Import all the negative samples collected by soft cascades, and split positive and negative samples into N sets $S = \{s_1, \ldots\ldots, s_N\}$.

1: For i = 1, ……, N

2: For t=1, ……, T

3: For n=1, ……, nWeak

4: Select subset of feature from pool of sample set s_i for training current weak classifier

5: Select the best classifier which has the minimum error. The error is defined as:

$$\epsilon_j = \sum_t \omega_i |h_j(x_i) - y_i|$$

6: Compute current weak classifier weight

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

7: Update samples' weights:

$$\omega_{t,i} = \frac{\omega_{t-1,i}}{\sum_j \omega_{t-1,j}}, \omega_{t-1,i} = w_{t-1,i}\beta_{t-1}^{1-\epsilon_i}$$

8: Return strong classifier of stage i:

$$H_i(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$$

The training algorithm of hard cascade classifier is described in Algorithm 1.

3 Description of Soft Cascade Classifier

Soft cascades classifier is shown in figure 3. Soft cascade classifier has a long stage which is constructed by many weak classifiers. In soft cascade classifier, one sample can be judged as pedestrian region only when it passes all of the weak classifiers. If any one of weak classifiers judge the sample as a negative one, the judging process is early stopped.

In the soft cascade classifier, each weak classifier is trained independently, which means we choose a subset training samples from the training set randomly with replacement. Compared with hard cascade classifier, the soft cascade classifier can reject background samples faster.

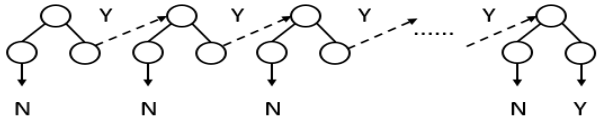

Figure 3: Workflow of soft cascade classifier

Algorithm 2 Training process of soft cascade detector

Input:

1. Training data set $\{(x_1, y_1), \ldots\ldots, (x_{a+b}, y_{a+b})\}$, a,b is the number of positive and negative samples, respectively.

Initialize:

1. 1.$\omega_{0,y_i} \leftarrow \frac{1}{2a}, \frac{1}{2b}$, initialize the weights of positive and negative samples, in which $y_i = 0,1$.

Train:

1: For t=1, ……, T

2: Collect m negative samples for training set with bootstrapping, and re-initialize samples' weights:

$$\omega_{0,y_i} \leftarrow \frac{1}{2a}, \frac{1}{2*(b+m)}$$

3: Randomly select feature subset s from feature pool which is generated by all training samples

4: Choose the best feature from s, which has the minimum classification error. The error is defined as:

$$\epsilon_j = \sum_i \omega_i |h_j(x_i) - y_i|$$

5: Define parameters to be updated:

$$\beta_t = \frac{\epsilon_t}{1-\epsilon_t}, \alpha_t = -log\beta_t, c_t = \alpha_t h_t$$

6: Update samples' weights:

$$\omega_{t,i} = \frac{\omega_{t-1,i}}{\sum_j \omega_{t-1,j}}, \omega_{t-1,i} = \omega_{t-1,i}\beta_{t-1}^{1-\epsilon_i}$$

7: return current stage sub-classifier c_t.

The training process of soft cascade classifier is described in Algorithm 2.

4 Combined Cascade Classifiers

As described in section 2 and section 3, hard cascade detection has better detection accuracy and soft cascade detection

can filter out background region faster. Thus, we design an algorithm to combine hard cascade and soft cascade which keep a high detection performance with faster detection rate. The combination of hard cascade and soft cascade is shown in Figure 3.

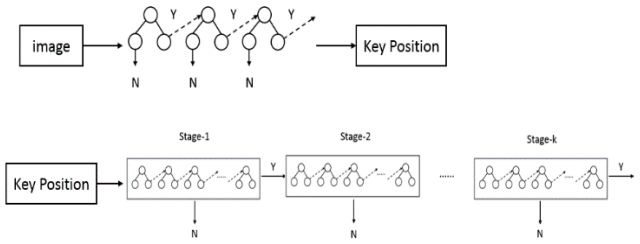

Figure 3: Combined cascade classifier

First, we use soft cascade classifier on the multi-scale image features to filter out background regions of an image. In this stage, we don't need a long cascade stage any more. Only a few weak classifier in soft cascade is enough. As experiments showed that a few weak classifiers in soft cascade can exclude most of the background regions. Thus, we design a method to compute the max length of soft cascade which can filter out the background regions. We define the method as Average Judging Trees of soft cascade.

With the average judging trees of soft cascade, many background regions are excluded and the image remains candidate regions that contain target pedestrians. We define the candidate regions as Key Positions. As these Key Positions contain target pedestrians, we use more powerful but slower hard cascade classifier to do finely detection. In a hard cascade classifier, each stage has more powerful ability on discriminating samples than the previous classifiers.

4.1 Average Judging Trees

Soft cascade consists of a long series of weak classifiers. A candidate region is judged as a positive one only when it passes all of the weak classifiers and meets all the thresholds of weak classifiers. On the otherwise, in an image, there are many background regions that means many regions need to be judged by only a few weak classifiers.

In our algorithm, to obtain the key positions in an image, we use clipped soft cascade classifier to filter out none target regions, namely a short version soft cascade. We define Average Judging Trees to describe the shortest length of a soft cascade. The function of average judging trees is defined as follows:

$$\text{times}_j = \left[(W_j - w)/\,stride + 1\right] * \left[(H_j - h)/stride + 1\right] \quad (1)$$

$$\text{ave\_tree\_num} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{scale}\sum_{k=1}^{times_j} tree\_num_k}{\sum_{i=1}^{N}\sum_{j=1}^{scale} time_j} \quad (2)$$

In function (1), time_j means that in the jth feature scale of an image the times of the pedestrian detector compared. In function (2), the tree_num_k means that in the jth feature scale of an image the number of weak classifiers passed in the kth detecting window.

After the soft cascade with average judging trees, the key positions in an image are kept for the next stage. The key positions contain candidate pedestrian regions and a few background regions. As described in section 2, each stage in a hard cascade classifier has more discriminative ability to classify positive and negative samples. In order to obtain better performance, we mainly use stages in the back of a hard cascade classifier to do finer detection on key positions.

5 Experiments

5.1 Dataset

We use INRIA Person Dataset[Dalal and Triggs, 2005] as train set / test set. In the train set, there are 614 images including 2416 pedestrians and 1218 images without any pedestrian. In the test set, there are 288 images including 1126 pedestrians and 453 images without any pedestrian. These images are collected from GRAZ-01 dataset, google images and some personal photos. Thus, these images are in high quality.

5.2 Description of Training Process

The training process of combined cascade classifier is shown in Figure 4. In the experiments, we use bootstrapping method to train hard cascade classifier and soft cascade classifier.

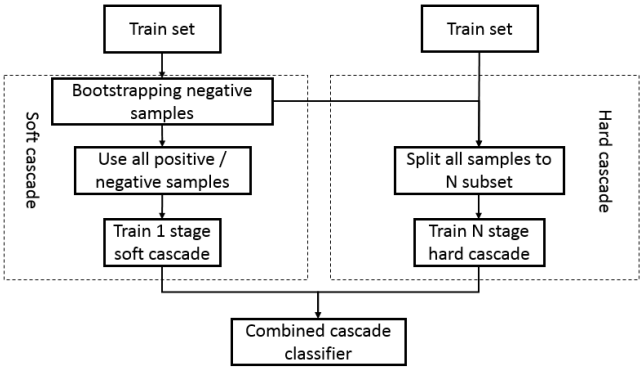

Figure 4: Training process of combined cascade classifier.

First, we train four stages of soft cascade classifier, in which the number of weak classifiers in each stage are incremental. The number of weak classifiers in each of four hard cascade stages is [32, 128, 512, 1024]. We use this training method for hard samples mining. Hard negative samples in current stage are all collected by last soft cascade classifier. After finishing soft cascade training, we keep only the last stage 1024 weak classifiers.

Second, we train six stages of hard cascade classifier, in which the number of weak classifiers is set to [16, 32, 64, 128, 256, 512]. We number the stages as ①~⑥. The train set is formed by positive samples and negative samples in INRIA dataset and by hard negative samples collected by the four-stage soft cascades. Thus, in the process of training hard cascade classifier, we use all hard negative samples that are bootstrapped by soft cascade classifier. We split these all train samples into six parts for training six hard cascade stages. And after finishing hard cascade training process, we keep all these stage classifiers.

5.3 Combination of Cascades

First, we detect on train set with the soft cascade to compute average judging trees. The experiment shows that on INRIA dataset the number of average judging trees is nine, which means we need only nine weak classifiers in soft cascade on average.

Last, we compare different combination methods on hard cascade classifier. The parts of combination of cascades' ROC curves are shown in Figure 5. The experiments show that the final cascade classifier combined by nine weak classifiers in soft cascade and ②, ⑥ stages in hard cascade has the best performance. Meanwhile, this method of classifier combination use less weak classifiers than both soft cascade and hard cascade.

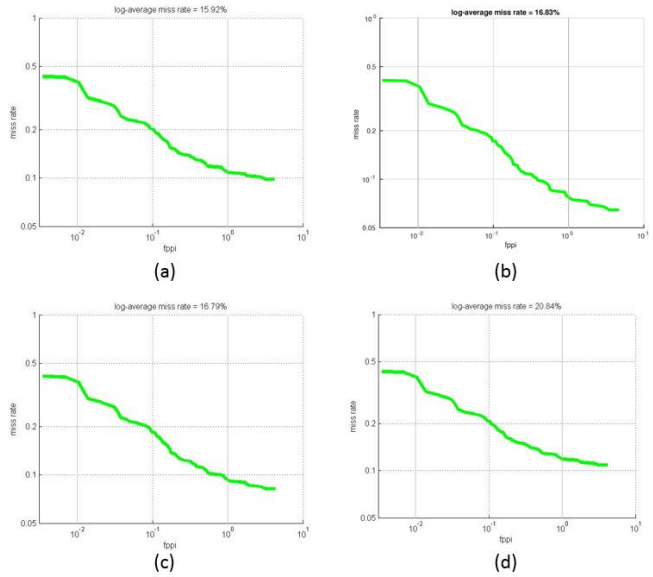

Figure 5: (a) ROC of hard cascade; (b) ROC of soft cascade; (c) ROC of nine weak classifiers in soft cascade and number ②, ⑥ stages in hard cascade; (d) ROC of nine weak classifiers in soft cascade and number ④.

In Figure 6, we compare the speed of different combination method. The 'S' in X axis means classifier with soft cascade, and the 'H' means classifier with hard cascade. The classifier consist of nine weak classifier in soft cascade and the six stages in hard cascade achieves the best result 37fps by a single core of Intel i5-6200U @ 2.3GHz CPU.

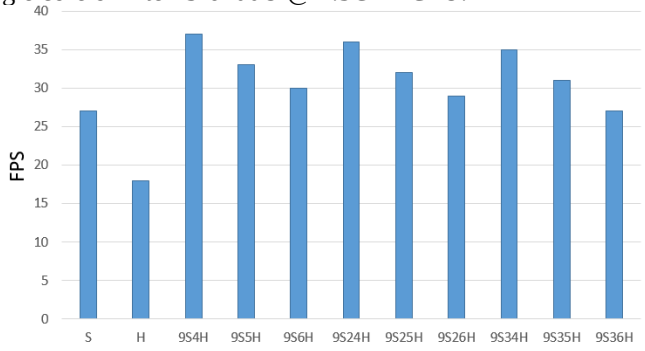

Figure 6: Different combination of soft and hard cascades.

6 Conclusion

In this paper, we propose a method of combining hard cascade and soft cascade for pedestrian detection. In order to filter out background in images faster and to maintain a high performance, we define a method of computing the shortest length of soft cascade and use a few parts of hard cascade to do fine pedestrian detection. The experiments show that with this method, the speed of pedestrian detection is highly improved and the accuracy of pedestrian detection is still in high performance.

References

[Felzenszwalb et al., 2013] Felzenszwalb P., Girshick R., Mcallester D. Visual object detection with deformable part models. *Communications of the ACM*, 2013, 56(9):97.

[Pedersoli et al., 2011] Pedersoli M , Vedaldi A , Gonzalez J . A coarse-to-fine approach for fast deformable object detection. *IEEE Conference on Computer Vision & Pattern Recognition*, 2011.

[Lampert et al., 2009] Lampert C H , Blaschko M B , Hofmann T. Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. *IEEE Transactions on Software Engineering*, 2009, 31(12):2129-2142.

[Dollár et al., 2012] Dollár P, Appel R, KienzleW. Crosstalk Cascades for Frame-rate Pedestrian Detection. *European Conference on Computer Visio*. Springer Berlin Heidelberg, 2012.

[Redmon et al., 2016] Redmon J , Divvala S , Girshick R , et al. You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2016.

[Redmon and Farhadi, 2017] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger. *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2017.

[Girshick, 2016] Girshick R. Fast R-CNN. *IEEE International Conference on Computer Vision*, 2016.

[Ren et al., 2015] Ren S, He K , Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2015, 39(6):1137-1149.

[Bourdev and Brandt, 2005] Bourdev L, Brandt J. Robust object detection via soft cascade. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[Viola and Jones, 2001] P. Viola and M. Jones. Robust real-time object detection. *IEEE International Conference on Computer Vision Workshop on Statistical and Computational Theories of Vision*, 2001.

[Dalal and Triggs, 2005] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. *IEEE International Conference on Computer Vision*, 2005.