Caltech 보행자 감지를위한 Scale-aware Faster R-CNN

바트후, 주마벡, 조근식
인하대학교 컴퓨터정보공학과
e-mail: bybatkhuu@yahoo.com

Scale-aware Faster R-CNN for Caltech Pedestrian Detection

Batkhuu Byambajav, Jumabek Alikhanov, Geun-Sik Jo
Dept. of Computer Science & Information Engineering, Inha University

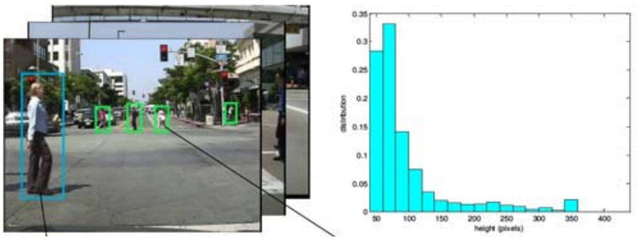
Abstract

We present real-time pedestrian detection that exploit accuracy of Faster R-CNN network. Faster R-CNN has shown to success at PASCAL VOC multi-object detection tasks, and their ability to operate on raw pixel input without the need to design special features is very engaging. Therefore, in this work we apply and adjust Faster R-CNN to single object detection, which is pedestrian detection. The drawback of Faster R-CNN is its failure when object size is small. Previously, small sized object problem was solved by Scale-aware Network. We incorporate Scale-aware Network to Faster R-CNN. This made our method Scale-aware Faster R-CNN (DF R-CNN) that is both fast and very accurate. We separated Faster R-CNN networks into two sub-network, that is one for large-size objects and another one for small-size objects. The resulting approach achieves a 28.3% average miss rate on the Caltech Pedestrian detection benchmark, which is competitive with the other best reported results.

1. Introduction

Pedestrian detection has been an important problem for last few decades and also future, given its applicability to a number of applications in robotics, autonomous driving, including driver assistance systems, road scene understanding or surveillance systems. It has attracted much attention within the computer vision field. The two main practical requirements are high accuracy and real-time speed: we need pedestrian detectors that are accurate enough and fast enough to run on systems with limited compute resource.

Pedestrian detection methods have employed a variety of techniques and features. Some have focused on increasing the speed of detection, whereas others have focused on accuracy. Recently, many research works on pedestrian detection and a novel range of methods have emerged, based on Deep Neural Networks, showing impressive accuracy gains [3]. However, they generally leave critical issues caused by many kind of scales of pedestrians in an image unsolved, which is shown to considerably affect the performance of pedestrian detection in natural scenes, and Deep Neural Network (DNN) models are known to be very slow especially when used as sliding-window classifiers. Statistically, over 60% of the instances from the Caltech [6] training set have a height smaller than 100 pixels. Accurately localizing these small-size pedestrian is quite challenging due to the following difficulties. First, most of the small-size instances are blurred boundaries and obscure appearance. It is difficult to distinguish them from the background clutters and other overlapped instances. Secondly, the large-size pedestrian instances typically exhibit dramatically different visual characteristics from the small-size ones. For instance, large-size body instances can provide rich information for pedestrian detection when small-size body instances are not easy to recognizable.



(a) Example Pedestrian Images

(b) Height Distribution on Caltech Training Set

Fig. 1. (a) Shows some example pedestrian, (b) Shows the distribution of pedestrians' heights on the Caltech training set.

Existing works address the scale-variance problem mainly from two aspects. First, the brute-force data augmentation (e.g., multi-scaling [2] or resizing [12]) is used to improve the scale-invariance capability. Second, single model with multi-scale filters is applied on all instances with various sizes. Based on above idea, we incorporated scale-aware network to our method, which is adjusted on for Faster R-CNN [1]. Faster R-CNN itself very fast, which is suitable in real-time detection but accuracy is lower than state-of-the-art methods. And we propose a very simple Scale-aware Faster R-CNN approach for pedestrian detection that is very accurate and runs in real-time. To achieve this, we separated Faster R-CNN network to two sub-networks, which is based on VGG16 [5] and ZF [4] networks. And added some specific layers. Adding convolutional layers will increase accuracy but computation would be expensive as much as many convolutional layers. In that case, we will add max pool layer to decrease computation size. If we increase max pool layers too much, it will be destructive to data features.

More specifically, we use the "Caffe" [10] open source implementation provided by Berkeley Vision and Learning Center and collaborators of the python Faster R-CNN algorithm. Our work makes the following contributions Firstly, we propose Scale-aware Faster R-CNN for pedestrian detection by large-size sub-network and small-size sub-network into unified architecture, following divide-and-conquer philosophy to increase detection accuracy. Secondly, Scale-aware Faster R-CNN is proposed to increase accuracy in classification for the final detection performance. Thirdly, experiments on challenging pedestrian dataset demonstrate that Scale-aware Faster R-CNN delivers new method in Caltech pedestrian benchmarks.

2. Related Work

Pedestrian detection has been one of the main topic in computer vision research area, more than 20 years of research. Many kind of methods have been used to pedestrian detection over the years, with continued improvement in performance. Some methods focus on improving the base features used, whereas others focus on the learning algorithms, or other techniques such as incorporating Deformable Parts Models [13]. Benenson et al. [7] have recently proposed a comparative paper that evaluates performance of various features and methods on pedestrian detection.

Viola and Jones proposed a cascade-of-classifiers approach [14], which has been widely used for real-time applications. The method has been extended by employing different types of features and techniques, but fundamentally the concept of the cascade, with early rejection of majority of test examples, has been widely utilized to achieve real-time performance. Perhaps the most popular feature used for pedestrian detection (and several other image-based detection tasks) is the HOG feature developed by Dalal and Triggs [15]. Although not real-time, about 1 FPS, this work has been instrumental to the development of faster and more accurate features for pedestrian detection, which are used in the top performing methods in combination with SVM or Decision forests. Deformable Parts Models [13] have shown success on the pedestrian detection task. Deep learning-based techniques have also been applied to pedestrian detection and have led to improvements in accuracy. These approaches are still slow, ranging from over a second per image to several minutes. The faster approaches do not apply deep nets to the raw pixel input so their accuracy is reduced. Improving the speed of pedestrian detection has also been an active area. Benenson et al. proposed a method reaching speeds of 100 to 135 FPS [16] for detection in a 480x640 image, albeit with significantly lower accuracy. Other researchers have focused specifically on speeding up Deep Neural Networks [12].

Object Proposals: There is a large literature on object proposal methods. Comprehensive surveys and comparisons of object proposal methods can be found where super-pixels (e.g., Selective Search [17], CPMC, MCG) and those based on sliding windows (e.g., objectness in windows, EdgeBoxes [18]). Object proposal methods were adopted as external modules independent of the detectors (e.g., Selective Search [17] object detectors, R-CNN, and Fast R-CNN [2]).

Deep Networks for Object Detection: The R-CNN method trains CNNs end-to-end to classify the proposal regions into object categories or background. R-CNN mainly plays as a classifier, and it does not predict object bounds (except for refining by bounding box regression). Its accuracy depends on the performance of the region proposal module.

3. Scale-aware Faster R-CNN

The architecture of the baseline deep neural network is based on the original deep network of VGG Net et al. [5] which has been widely adopted and used by many researchers.

However, it is very slow when ran in a sliding window fashion. One key difference here is that we reduced the depths of some of the convolutional layers and the sizes of the receptive fields, which is specifically done to gain speed advantage. Even with the proposed speedups by other researchers, this network is still slow and not appropriate for real-time applications. To speed it up we utilize the idea of a small convolutional network from ZF network work. When ran in a ZF net, it will process all image patches first, and pass through only the patches that have high confidence values. It reduces the massive computational time that is needed for a sliding window detector to evaluate a full DNN at all candidate locations and scales. The combination of these two deep networks works 80 times faster than the original sliding window. This speedup is also a crucial component in achieving real-time performance in our method, described below.

3.1. Architecture of Scale-aware Faster R-CNN

Figure 2 illustrates the architecture of Scale-aware Faster R-CNN in details. The Scale-aware Faster R-CNN passes the input image into several convolutional layers and max pooling layers to extract feature maps. Then proposed network separate into two sub-networks, which are learned to detect large-size and small-size object separately. Each of the two sub-networks takes as input the feature maps produced from the previous convolutional layers, and further extracts features through several convolutional layers to produce feature maps specialized for a specific range of input scales.

3.2. Runtime

We measure the runtime on a standard NVIDIA K40 Tesla GPU. The Scale-aware Faster R-CNN takes about 67 milliseconds (ms) to detection. About 64 patches are passed through per image from the input. The first stage of the Scale-aware Faster R-CNN runs at 17 ms per batch of 128. The second stage of the Scale-aware Faster R-CNN (which is the baseline classifier) takes about 50ms. The overall runtime is about 67ms per 640x480 image, which is 15 frames per second. Previous methods, e.g. Luo et al. [19] have similarly employed a hybrid approach, HOG-based cascade and a deep network at the bottom of the cascade, but their runtime is about 1-1.5 seconds also on GPU, which is 21 times slower.

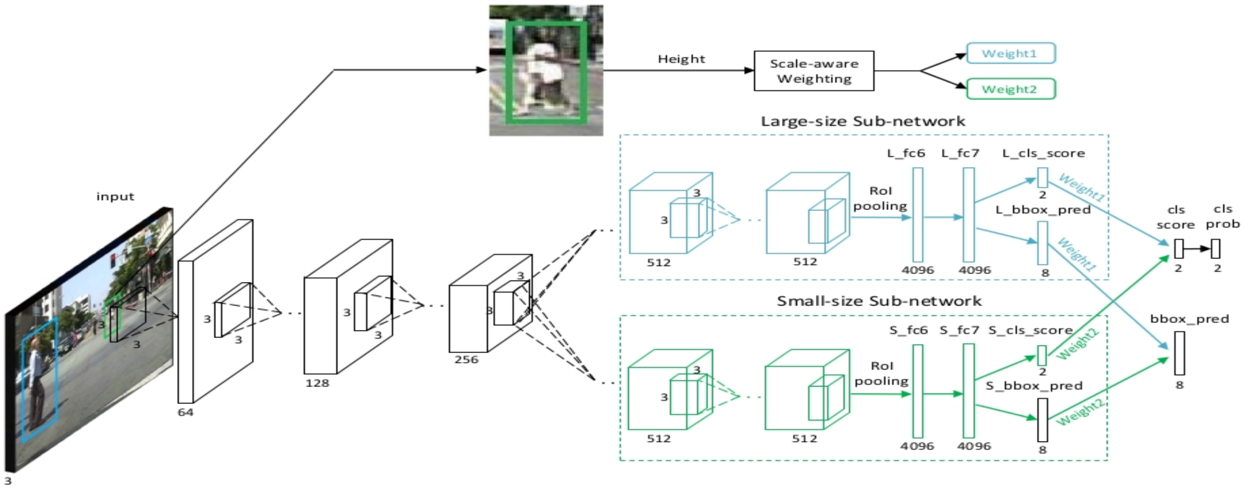


Fig. 2. The architecture of our Scale-aware Faster R-CNN

3.3. Implementation

Pretraining: We make use of pre-training, that is, the weights are initialized from the weights of a network that has been trained on ImageNet and Pascal VOC. Other works have noted similar effects, and since pre-training is easy to incorporate, it is a preferred choice in our work and others.

Data generation: A standard procedure for data generation is used, in which we crop a square box around pedestrian examples. At the time of data generation, the cropped square images are resized to 72x72. Additional random crops of size 64x64, to match the input size of the network, are taken at each iteration during training of the DNN. This is a standard data augmentation technique for training convolutional networks and allows more diverse ‘views’ of the training examples. The small network follows the same procedure, with an additional resizing of the input. We further collect hard negatives, which is important because the initial generated dataset is sampled uniformly from the available examples, and contains a large fraction of easy examples. Additionally, we eliminated the pedestrian examples that are smaller than 10 pixels in width, as these examples are indistinguishable when seen as individual patch and are not useful for methods that do not apply motion (as is ours).

4. Experiment

Datasets

We evaluated our results on the standard Caltech Pedestrian detection dataset, as this dataset serves as a standard benchmark for pedestrian detection methods. We further consider additional pedestrian datasets. Details are below.

Caltech dataset: The Caltech dataset contains about 50,000 labeled pedestrians. The dataset is collected from a dashboard color camera and contains suburban and city scenes. In our experiments we used about two thirds of the available pedestrians as some of them are of very small sizes and poor quality.

Independent pedestrian dataset: We independently collected pedestrian dataset which is comparable in size to the Caltech one. It contains higher quality images of pedestrians due to the better resolution of the cameras used. We wanted to measure performance when not training on the Caltech data, but on an independent dataset of similar size.

Extra pedestrian dataset: We complemented the Caltech training set with additional examples to further experiment with the effects of adding more data. This dataset consists of the Caltech training dataset as above, plus the publicly available ETH and Daimler pedestrian datasets. The Daimler dataset contains only grayscale images and the ETH dataset is collected from a mobile platform, in this case a stroller.

Evaluation of the Scale-aware Faster R-CNN

In our evaluation, we use the training and test protocols established in the Caltech pedestrian benchmark and report the results by measuring the average miss rate as prior methods did. We use the code provided in the toolbox of this benchmark to do the evaluation. We first evaluate the performance of the Scale-aware Faster R-CNN as described in Section 3. Our results are in Table 1, listed with current state-of-the-art pedestrian detection methods. We tested on pedestrians of at least 50 pixels (’reasonable set’). Our Scale-aware Faster R-CNN trained on Caltech by fune-tuning in Pascal VOC dataset. Compared to state-of-the-arts, our methods with average miss rates of 28.3%, outperform most approaches, including all deep learning-based ones.

The only exception are the approaches that use additional motion features SpatialPooling+Katamari, which perform at 22%. The Deep Faster R-CNN approach, is outperformed by the SpatialPooling method that directly optimizes the area under the curve. The SpatialPooling method performs better than most other methods for very high false positive values, and this part of the curve is not useful for practical applications.

Table 1. Summary of results of the Scale-aware Faster R-CNN methods compared with state-of-the-art.

Method	Avg. miss rate	FPS
MOCO [20]	45.5	1
MT-DPM+Context [13]	37.64	16
Hosang et al. [21]	32.4	0.63
SpatialPooling [22]	29.0	0.13
SpatialPooling+/Katamari (w. motion)	22.0	0.13
DeepCascade [23]	31.11	15
DeepCascadeID w. extra data [23]	26.21	15
Scale-aware Faster R-CNN	28.3	15

5. Conclusion

We have presented Deep Neural Network-based algorithm for pedestrian detection which adjusts Faster R-CNN for pedestrian detection. We also incorporate the idea of Scale-aware Network to Faster R-CNN to improve our performance. This involved to separate Faster R-CNN network into two different sub-networks which are responsible for small-sized and large-sized pedestrians. That makes our method very efficient. We also applied multiple class object detection method to only one class object detection, which is pedestrian detection. Our method ranks method the best ones for pedestrian detection and runs in real-time, 15 frames per second. This shows Faster RCNN is efficiency not only multi class object detection but also for pedestrian detection. Therefore, in future could be applied for other single object detection tasks.

We hope this method will help future works to continue to improve pedestrian detectors in terms of both accuracy and speed so that more methods can be usable for pedestrian detection for real-time applications. Future work can include increasing the depth of the Scale-aware Faster R-CNN by adding tinier deep networks and exploring the efficiency-accuracy trade-offs. We further plan to explore using motion information from the images, because clearly motion cues are extremely important for such applications.

References

[1] S. Ren, K. He, R. Girshick, J. Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks”, in Neural Information Processing Systems (NIPS), 2015.

[2] R. Girshick. “Fast R-CNN”, in IEEE International Conference on Computer Vision (ICCV), 2015.

[3] A. Krizhevsky, I. Sutskever, G. Hinton. “Imagenet classification with deep convolutional neural networks”, in Neural Information Processing Systems (NIPS), 2012.

[4] M. D. Zeiler, R. Fergus. “Visualizing and understanding convolutional neural networks”, European Conference on Computer Vision (ECCV), 2014.

[5] K. Simonyan, A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”, 2015.

[6] P. Dollar, C. Wojek, B. Schiele, P. Perona. “Pedestrain Detection: An Evaluation of the State of the Art”, 2012.

[7] S. Zhang, R. Benenson, M. Omran, J. Hosang, B. Schiele. “How Far Are We from Solving Pedestrian Detection?”, 2016.

[8] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, S. Yan. “Scale-aware Fast R-CNN for Pedestrian Detection”, 2016.

[9] M. Everingham, L. Van Gool, Christopher K. I. Williams, J. Winn, A. Zisserman. “The Pascal Visual Object Classes (VOC) Challenge”, 2010.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell. “Caffe: Convolutional architecture for fast feature embedding”, arXiv:1408.5093, 2014.

[11] N. Dalal, B. Triggs. “Histograms of oriented gradients for human detection”, in CVPR, pages 886–893, 2005.

[12] R. Girshick, J. Donahue, T. Darrell, J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”, In CVPR, pages 580–587, 2014.

[13] J. Yan, X. Zhang, Z. Lei, S.Liao, S. Li, “Robust multi-resolution pedestrian detection in traffic scenes”, CVPR, 2013

[14] P. Viola, M. Jones, D. Snow, “Detecting pedestrians using patterns of motion and appearance”, ICCV, 2003.

[15] N. Dalal, B. Triggs, “Histograms of oriented gradients for human detection”, CVPR, 2005.

[16] R. Benenson, M. Matthias, R. Tomofte, L. Van Gool, “Pedestrian detection at 100 frames per second”, CVPR, 2012.

[17] J. R. Uijlings, K. E. van de Sande, T. Gevers, A. W. Smeulders, “Selective search for object recognition”, International Journal of Computer Vision (IJCV), 2013.

[18] C. L. Zitnick and P. Doll´ar, “Edge boxes: Locating object proposals from edges”, European Conference on Computer Vision (ECCV), 2014.

[19] C. Szegedy, A. Toshev, D. Erhan, “Deep neural networks for object detection”, Neural Information Processing Systems (NIPS), 2013.

[20] G. Chen, Y. Ding, J. Xiao, T. Han, "Detection evolution with multi-order contextual co-occurrence", CVPR, 2013.

[21] J. Hosang, M. Omran, R. Benenson, B. Schiele, "Taking a deeper look at pedestrians", CVPR, 2015.

[22] S. Paisitkriangkrai, C. Shen, A. van den Hengel, "Strengthening the effectiveness of pedestrian detection", ECCV, 2014.

[23] A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, D. Ferguson, “Real-Time Pedestrian Detection With Deep Network Cascades”, 2015