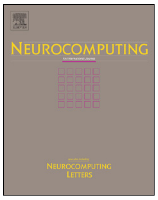
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



# Speed up deep neural network based pedestrian detection by sharing features across multi-scale models

Xiaoheng Jiang [a], Yanwei Pang [a,*], Xuelong Li [b], Jing Pan [a,c]

[a] School of Electronic Information Engineering, Tianjin University, Tianjin 300072, PR China
[b] Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, PR China
[c] School of Electronic Engineering, Tianjin University of Technology and Education, Tianjin 300222, PR China

ARTICLE INFO

*Article history:*
Received 11 September 2015
Received in revised form
18 November 2015
Accepted 16 December 2015
Communicated by Feiping Nie

*Keywords:*
Pedestrian detection
Deep neural networks
Convolutional neural networks
Share features

ABSTRACT

Deep neural networks (DNNs) have now demonstrated state-of-the-art detection performance on pedestrian datasets. However, because of their high computational complexity, detection efficiency is still a frustrating problem even with the help of Graphics Processing Units (GPUs). To improve detection efficiency, this paper proposes to share features across a group of DNNs that correspond to pedestrian models of different sizes. By sharing features, the computational burden for extracting features from an image pyramid can be significantly reduced. Simultaneously, we can detect pedestrians of several different scales on one single layer of an image pyramid. Furthermore, the improvement of detection efficiency is achieved with negligible loss of detection accuracy. Experimental results demonstrate the robustness and efficiency of the proposed algorithm.

## 1. Introduction

The past few years have witnessed the successful application of deep neural networks (DNNs) in the field of computer vision [1,6,11,12,15,17,19,28,34]. Especially, convolutional neural networks (ConvNets) have attracted much attention on object classification and object detection [2,7,9,16,20,25,29]. To testify the effectiveness of DNNs, pedestrian detection (a canonical instance of object detection) [3,4,24] inevitably becomes the focus since pedestrian usually presents a wide variety of appearances due to body pose, occlusions, clothing, lighting, and backgrounds. Furthermore, pedestrian detection has direct applications in car safety, surveillance, and robotics.

Although several previous works on DNNs for pedestrian detection have achieved promising performance, they paid more attention to improve the detection accuracy rather than the efficiency. Typically, a trained DNN detector takes as input one fixed-size image patch, which is obtained by means of sliding window technique [13,26,27] or region proposal methods [7]. The high computational cost of the DNN detector makes sliding window method unattractive because of the resulting countless image patches. Region proposal methods [31,33] can generate a reduced set of image patches, usually two orders of magnitude fewer compared to the sliding window approach. However, the recall of region proposal methods drops sharply with the increase of Intersection-over-Union (IoU) threshold, thus significantly decreasing the detection accuracy. An alternative is to use dynamic programming to speed up DNN detectors [8]. Dynamic programming method differs from sliding window approach in that it extracts image-level features once and for all. The final decision is made at the top layer (i.e., the last feature layer) in sliding window fashion. Image-level scanning avoids the heavy computation wasted at overlapping regions between neighbouring image patches, thus speeding up the detection process by orders of magnitude.

Even by adopting image-level scanning, it is necessary to construct an image pyramid with dozens of layers. In this paper, based on image-level scanning, we propose to share features across a group of DNN detectors that correspond to pedestrian models of different sizes. The final decision made at the top layer of the shared features can simultaneously detect pedestrians of several different scales. This strategy can reduce the number of image pyramid layers that are needed for extracting features and thus relieve computational burden. Note that in this paper, we focus on speeding up the DNN based pedestrian detectors while maintaining the detection accuracy. A comprehensive comparison of pedestrian detectors with different DNN architectures is beyond the scope of this paper.

[*] Corresponding author.
*E-mail addresses:* jiangxiaoheng@tju.edu.cn (X. Jiang), pyw@tju.edu.cn (Y. Pang), xuelong_li@opt.ac.cn (X. Li), jingpan23@gmail.com (J. Pan).

http://dx.doi.org/10.1016/j.neucom.2015.12.042

In summary, the main contribution of the paper is as follows. (1) We propose to share features across a group of DNNs corresponding to pedestrian models of different sizes. (2) We propose to simultaneously detect pedestrians of several different scales on one layer of an image pyramid. The rest of the paper is organized as follows. Section 2 reviews some previous works on DNNs. Section 3 presents the image-level scanning method when exploiting DNN based object detectors. Our method is presented in detail in Section 4. Experimental results are reported in Section 5. Finally, conclusions are presented in Section 6.

## 2. Related work

The great success of ConvNets on the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [15] aroused broader interest in DNNs among researchers. Other than classification task, DNNs have also been used for detection task on ImageNet and Pascal VOC categories. The most successful generic object detectors are based on variants of the R-CNN framework [7]. In the R-CNN framework, selective search method [31] is utilized to first generate a relatively small set of detection proposals from the input image, and then these proposals are evaluated via a ConvNet.

However, when it comes to works on DNNs for pedestrian detection, few uses generic methods (e.g., selective search [31] and edge boxes [33]) to generate detection proposals. As pointed out in [10], a sophisticated pedestrian detector such as SquaresChnFtrs [2] can generate much better proposals than the state-of-the-art generic methods. Taking a deeper insight into generic methods, it is observed that the recall of ground truth annotations drops sharply when the IoU threshold increases. A low recall means that the generic methods would reject the majority of pedestrian instances in advance, resulting in a high miss rate even though the pedestrian detector performs very well. On the other hand, as the object detector is sensitive to object location [30], a coarse proposal location (i.e., a proposal with low IoU value) is likely to result in a low decision score. A low score also leads to losing one potential pedestrian instance, thus increasing the miss rate. In other words, the current generic proposal methods fail to generate detection proposals that are good enough for pedestrian detection.

Instead of using generic proposal methods, DNN based pedestrian detection works typically adopt another two approaches to generate detection proposals. Sermanet et al. [26] designed a ConvNet based pedestrian detector by combining features from the last two layers for detection. This ConvNet detector is then applied in sliding window fashion during detection stage, with a scale stride of 1.10 between each scale. A different line of work utilizes existing sophisticated detectors to generate detection proposals. In [21], Ouyang et al. proposed to construct a discriminative deep model with a stack of Restricted Boltzmann Machines (RBMs). This deep model extends classic deformable part model (DPM) [5] and is able to reason about pedestrian parts and occlusions. Later, Ouyang et al. [23] further extended the deep model to account for person-to-person relations. The above two works use DPM detector for proposals. In [22], Ouyang et al. incorporated feature extraction, deformation handling, occlusion handling, and classification into a new deep ConvNet and optimized the four components jointly. This new deep ConvNet utilizes a HOG+CSS+linear SVM detector [32] to generate proposals since it is of high computational complexity. The multi-stage contextual deep model (MSCD) [35] feeds each layer with contextual features computed at different scales around the candidate pedestrian detection. Switchable deep network (SDN) [18] improves ConvNet for pedestrian detection by adding multiple switchable layers built with a new switchable RBM. Both MSCD and SDN also use a HOG+CSS+linear SVM detector for proposals. Hosang et al. [10] used straightforward ConvNets (i.e., ConvNets without custom designs) such as the small CifarNet [14] and the big AlexNet [15] for pedestrian detection and adopted the SquaresChnFtrs detector [2] to generate proposals. Those sophisticated detectors mentioned above can generate good detection proposals for DNN based pedestrian detectors. However, they are in themselves very time consuming and take most of the detection time.

It is computationally prohibitive to scan an image with DNNs by means of sliding window, even utilizing GPU technique. The reason is that the neighbouring image patches are typically heavily overlapped and thus a significant amount of redundant computation is consumed. Luckily, it is pointed out in [8] that the scanning process of ConvNets can be speeded up by computing all convolutions in the first layer on the entire input image, and then computing all convolutions in subsequent layers on the resulting extended maps. This kind of scanning which is called image-level scanning can avoid redundant computation consumed by patch-level scanning. In [30], by applying image-level scanning flexibly, the authors could adjust the scanning resolution in the input dimension along each axis. We will briefly review the image-level scanning method in the next section as it is the foundation of our approach. Readers can refer to [8] for fully detailed information.

## 3. Image-level scanning

A ConvNet typically owns convolutional and pooling layers. The convolutional layer generates output feature maps by convolving the input maps (input image or output feature maps of the last layer) with a stack of kernels. The pooling layer pools information of a given region on the output maps. Given an image patch of $32 \times 32$ pixels, one can construct a ConvNet that contains two convolutional layers, two pooling layers, and one fully-connected layer, as shown in Fig. 1. The two convolutional layers both have a kernel of size $5 \times 5$ with a stride of 1 pixel, and the two pooling layers both have a kernel of size $2 \times 2$ with a stride of 1 pixel. The final fully-connected layer is represented by a convolutional kernel of size $5 \times 5$, which is exactly the same size of the input map.

As shown in Fig. 1, the ConvNet produces a single spatial output during training state. When applied at test time over an image (much larger than $32 \times 32$) in patch-level scanning fashion with a stride of 4 pixels, for example, it produces a single spatial output every time it moves, as shown in Fig. 2.

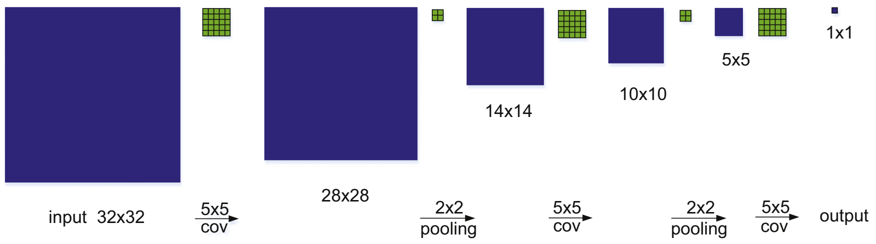


**Fig. 1.** An example of ConvNet.

Please cite this article as: X. Jiang, et al., Speed up deep neural network based pedestrian detection by sharing features across multi-scale models, Neurocomputing (2016), http://dx.doi.org/10.1016/j.neucom.2015.12.042