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
0925-2312/

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 × 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 × 5 with a stride of 1 pixel, and the two pooling layers both have a kernel of size 2 × 2 with a stride of 1 pixel. The final fully-connected layer is represented by a convolutional kernel of size 5 × 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 × 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.

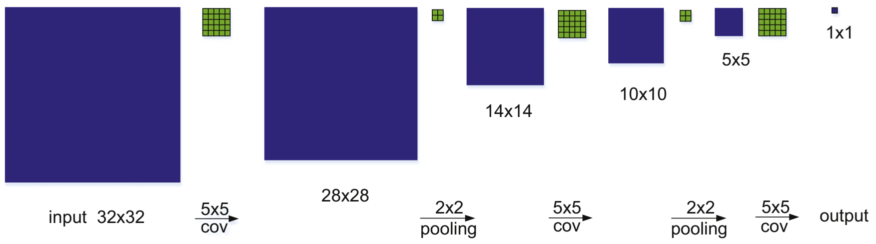


Fig. 1. An example of ConvNet.

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

Apparently, convolutions are re-evaluated on overlapping regions between adjacent image patches, consuming a huge amount of redundant computation. Instead, image-level scanning can avoid redundant convolution because it convolves the entire input maps at a time, as shown in Fig. 3. In Fig. 3, when applied at test time over a larger image (36 × 36) in image-level scanning way, the ConvNet only spends extra computation on the yellow regions. Therefore, it is better to compute each convolution only once for the whole input image and the resulting set of maps contains the maps for each patch contained in the input image. Moreover, each of the final 2 × 2 outputs corresponds to a single spatial output obtained in patch-level scanning shown in Fig. 2. This is because the two pooling layers, each of which has a kernel of size 2 × 2, result in a total sub-sampling ratio of 2 × 2 or 4 in the ConvNet described above. As a result, a stride of 1 pixel on the second pooling layer (output map of size 6 × 6) corresponds to a stride of 4 pixels on the input image. Therefore, in the case of ConvNets, image-level scanning approach inherently is similar to patch-level scanning, except for that it is much more efficient.

4. Our method

Pedestrian usually varies largely in size in a given image. At test time, two strategies are generally used to detect pedestrians of different sizes. One strategy is to keep the pedestrian model unchanged and iteratively shrink the image by a constant scaling factor (e.g. 1.25). On the contrary, the other strategy is to keep the input image unchanged and utilize a series of pedestrian models of different sizes. In this paper, based on image-level scanning, we propose to share features across a group of ConvNets corresponding to pedestrian models of different sizes and detect pedestrians of several different sizes on one layer of an image pyramid. The proposed method is characterized by traits of both strategies above. On the one hand, we also scale the input image, but with a much larger scaling factor. On the other hand, we use a group of pedestrian models of different sizes.

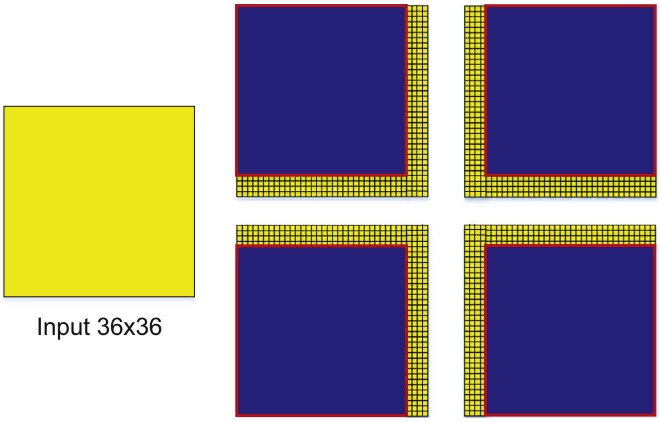


Fig. 2. Patch-level scanning.

4.1. Sharing features across multi-scale models

In [7], it is demonstrated that a large ConvNet (called as AlexNet) trained for whole-image classification on ImageNet works well for detecting objects in PASCAL. Furthermore, in [10], AlexNet shows promising performance on the task of detecting pedestrian, even without fine-tuning on pedestrian dataset. The success of transferring image representation of ConvNets from one task (or dataset) to another task (or dataset) is mainly attributed to their ability to learn a hierarchy of features. It should be noted that previous works focused on the transferring between different datasets. However, in this paper, we propose to share features across a group of ConvNet based pedestrian detectors with model windows of different sizes. This means that we concentrate on the transferring among models of different sizes on the same pedestrian dataset.

A typical choice for pedestrian detectors is a model window of 128 × 64 pixels, based on which we present our ConvNet detector in Fig. 4(a). We call it PedConv 1. PedConv 1 contains five layers with weights: the first four are convolutional and the last one is fully connected. The last layer (output layer) can be denoted in convolutional form with a kernel of size 14 × 6 × 128. This is because there are 128 output maps obtained from the fourth convolutional layer, and each of the output maps is of size 14 × 6 pixels.

Suppose there is a new ConvNet based pedestrian detector which has a model window of size 152 × 76 pixels. We now re-use the same parameters used in the first four layers of PedConv 1. As a result, the fourth layer outputs 128 maps of size 17 × 7, as shown in Fig. 4(b). These maps are different from those of size 14 × 6 in PedConv 1. Therefore, the last layer of Fig. 4(b) is represented by a kernel of size 17 × 7 × 128. We call this new ConvNet detector PedConv 2. PedConv 1 and PedConv 2 can be trained on one pedestrian dataset, respectively. As a result, the parameter values of PedConv 1 will be totally different from those of PedConv 2. If we run PedConv 1 and PedConv 2 on one input image in image-level scanning way, respectively, we have to repeat the convolutional operations in the first four layers because their convolutional kernels have different values.

To reduce redundant convolutional computation, we now intend to copy all the parameter values of the first four layers of PedConv 1 when training PedConv 2. That is, the parameter values of the convolutional layers of both PedConv 1 and PedConv 2 are the same. As a result, the only difference between PedConv 1 and PedConv 2 lies in the last layer. Now, if we run PedConv 1 and PedConv 2 on one input image in image-level scanning way, respectively, we only need to carry on convolutional computing in the first four layers once.

If there are a group of ConvNet based pedestrian detectors with model windows of different sizes, sharing features will result in a significant decrease in convolutional computation. The problem

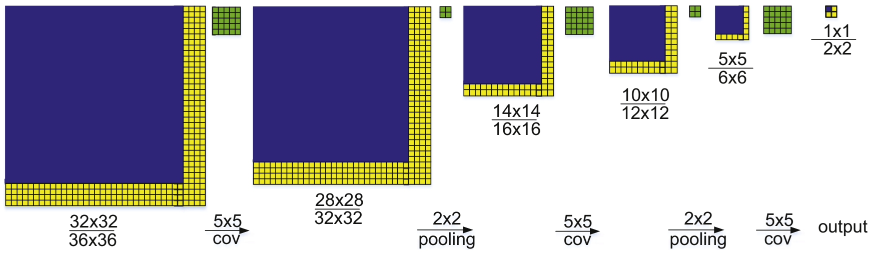


Fig. 3. Image-level scanning. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

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

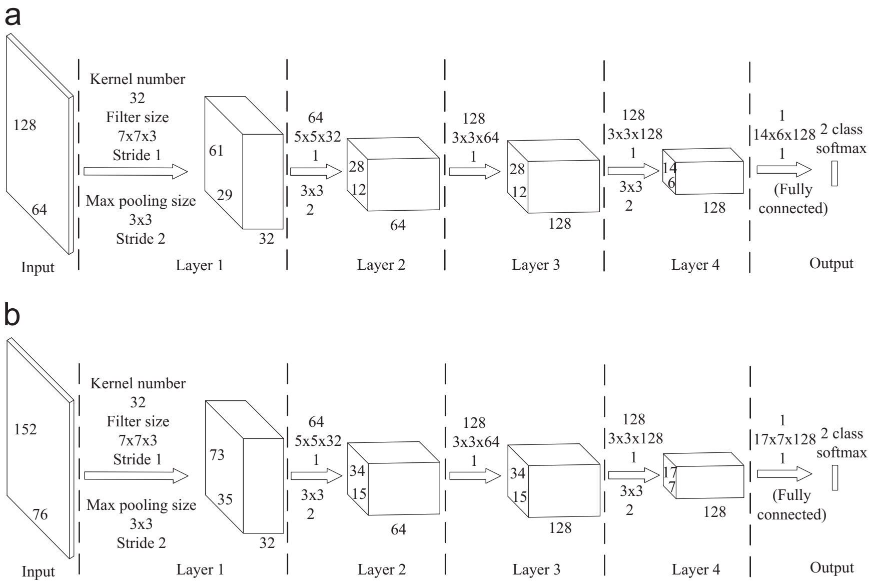


Fig. 4. ConvNet based pedestrian detectors with model windows of different sizes.

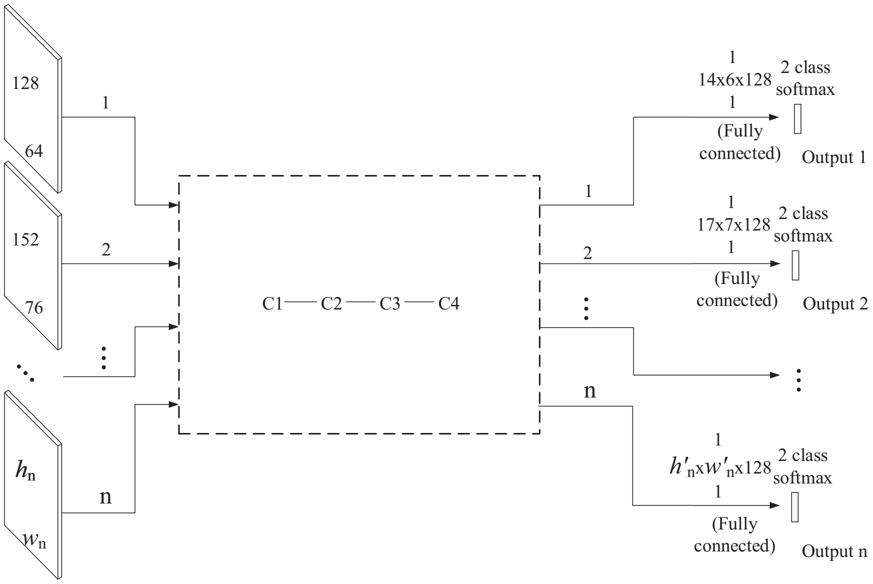


Fig. 5. Sharing features across a group of ConvNet detectors.

that arises is the feasibility of sharing these features across multi-scale models. Our answer is yes. ConvNet is characterized by its large capacity of learning a hierarchy of features. The lower-level feature layers are able to learn some common simple feature patterns among different objects, such as line, dot, and cross. Feature transferring between different datasets [7] has shown great success and it can benefit from the overlap between classes in source and target datasets. During our training stage, the source and target datasets are actually derived from the same pedestrian dataset, and they only differ in the size of model window (e.g., 128 × 64 vs. 152 × 76). Therefore, it is feasible to share features across multi-scale pedestrian models on the same dataset. The experimental results in next section verify our assumption. One illustration of sharing features across multi-scale pedestrian models is shown in Fig. 5. In Fig. 5, a group of pedestrian detectors with model window of different sizes share the features of the first four convolutional layers. The first four layers are denoted by C1, C2, C3, and C4, respectively.

4.2. Multi-scale detection on one single image pyramid layer

Once sharing features across multi-scale pedestrian models, it is feasible to detect pedestrians of different sizes on one single layer of an image pyramid. One illustration of multi-scale detection is shown in Fig. 6. Given one input image (or one layer of an image pyramid) of size $h \times w$, the final feature maps can be obtained by running the shared parts of different models on it for only once in image-level scanning fashion. The size of each feature map is $h' \times w'$. Suppose there are four ConvNet detectors with model windows of different sizes, then there will be four convolutional kernels that are carried out on the final feature maps accordingly. The four convolutional kernels are denoted by cuboids of different sizes and colors, as shown in Fig. 6. In fact, each cuboid on final the feature maps corresponds to an image patch of the input image.

One specific instance of multi-scale detection is presented in Fig. 7. For simplicity, we just use one final feature map. There are four ConvNet based pedestrian detectors, the sizes of model

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

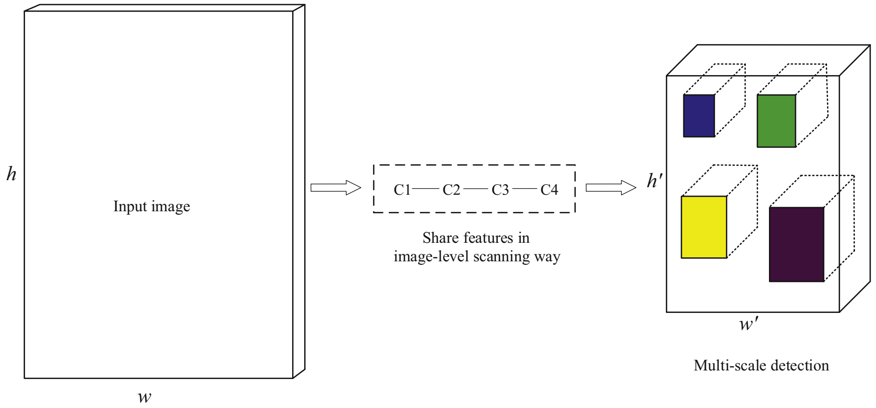


Fig. 6. Multi-scale detection on one layer of an image pyramid.

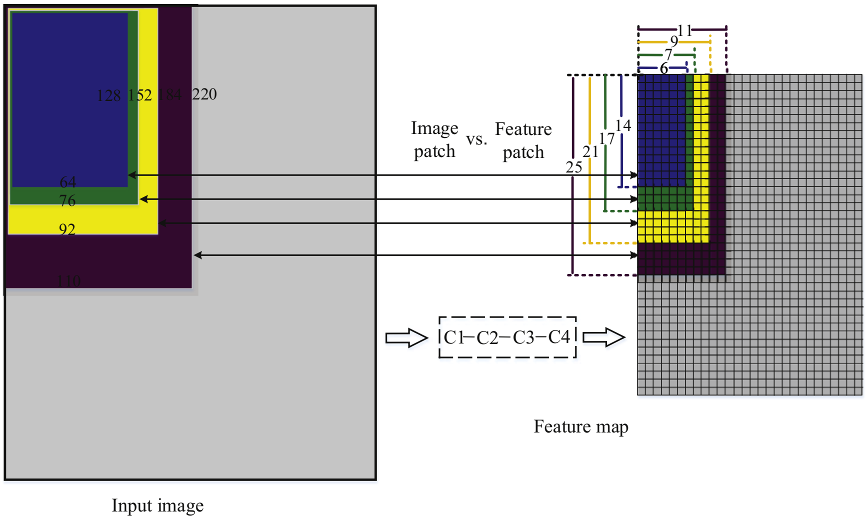


Fig. 7. One specific instance of multi-scale pedestrian detection on one single layer of an image pyramid.

windows of which are 128 × 64, 152 × 76, 184 × 92, and 220 × 110, respectively. On the final feature map, the sizes of the corresponding feature patches are 14 × 6, 17 × 7, 21 × 9, and 25 × 11, respectively. Suppose there are three pooling layers, each of which has a pooling region of size 2 × 2 and a stride of 1 pixel. Then, if the feature patch moves with a one-pixel stride in sliding window fashion, the corresponding image patch will move with a eight-pixel stride on the input image. As a result, sliding the four feature patches on the final feature map means that it is able to detect pedestrians of four different scales on the input image.

Unlike traditional sliding window approach which needs a fine or small scaling factor when generating an image pyramid, the proposed method only needs a coarse or much larger scaling factor. Suppose the scaling factor between each scale is s when using only one ConvNet detector, then the scaling factor s' when using n ConvNet detectors can be denoted by $s' = s^n$. As a result, the layers that are needed for extracting features can be largely reduced. In other words, this strategy can reduce the computation spent on extracting features.

5. Experiments

It should be noted that ConvNet based pedestrian detectors have achieved state-of-the-art performance on pedestrian datasets [10]. In this paper, we mainly focus on improving the efficiency of ConvNet based pedestrian detectors. Therefore, the comparison of different ConvNets is beyond our scope. The proposed method can be applied to any other ConvNet based object detectors. We evaluate our method on two standard pedestrian detection datasets: INRIA [3] and Caltech-USA [4]. However, we only train on the INRIA dataset because of its diversity. Caltech-USA is now the predominant benchmark for pedestrian detection as it is comparatively large and challenging. In addition, we do not pre-train the ConvNet detectors on any other dataset as detection performance is not our focus. It should be noted again that our purpose is to improve the detection efficiency of ConvNet based pedestrian detectors while maintaining their detection performance.

5.1. Data preparation

The ConvNet based pedestrian detector is trained on the INRIA pedestrian dataset. As pointed out in Section 4.1, we intend to train a group of ConvNet detectors that have model windows of different sizes. The basic ConvNet detector has a model window of size 128 × 64. The size of each model window increases gradually by a scaling factor 1.20 because it is typically adopted by other methods. Firstly, pedestrians are extracted into windows of 128 pixels in height and 64 pixels in width. Each pedestrian image is mirrored along the horizontal axis to expand the dataset. In addition, we add eight translation variations of both original and mirrored sample using a constant translation stride of two pixels, that is, from top left (−2, −2) to bottom right (2, 2). This results in a total of 21,744 pedestrian samples. An equal amount of background samples are extracted at random from negative images. We take 10% of all the extracted samples for validation, yielding a validation set with 4348 samples and a training set with 39140 samples. Secondly, we resize the samples of both validation set and training set in accordance with the sizes of other model windows. In our experiment, the smallest model window is of size 128 × 64. Therefore, in Caltech-USA test dataset, we only evaluate pedestrians that are bigger than 85 pixels in height.

5.2. Training

In our experiment, we utilize MatConvNet provided by VLFeat open source library. MatConvNet is a MATLAB toolbox implementing CNNs for computer vision applications. It is simple and efficient and can run and learn state-of-the-art CNNs. To begin the experiment, we firstly train a ConvNet based pedestrian detector with model window of size 128 × 64 and use it as the baseline

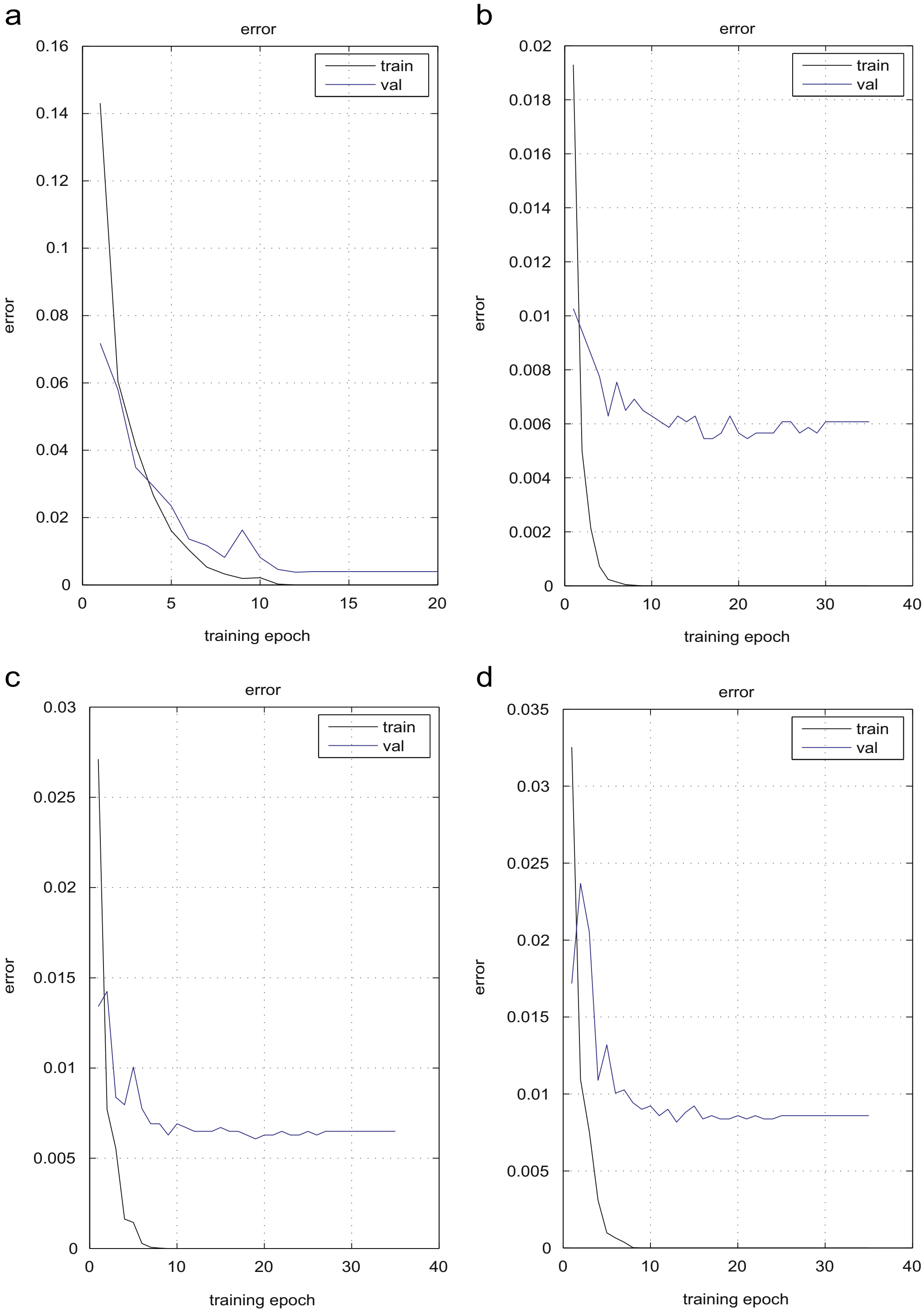


Fig. 8. Training results by sharing features. (a) Model window of size 128 × 64, end-end training. (b), (c) and (d) have model windows of sizes 152 × 76, 184 × 92 and 220 × 110, respectively, only training the last layer by sharing features learned from (a).

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

detector. We call it PedConv 1 as we do in Section 4.1. The structure of PedConv 1 is presented in Fig. 4(a). After obtaining PedConv 1, we obtain PedConv 2, PedConv 3, and PedConv 4 by just training the corresponding last convolutional layers, respectively. As depicted in Fig. 5, the last layers of PedConv 2, PedConv 3, and PedConv 4 have convolutional kernels of $17 \times 7 \times 128, 21 \times 9 \times 128$, and $25 \times 11 \times 128$, respectively.

The training results are presented in Fig. 8. It is seen that the validation error increases gradually from PedConv 1 to PedConv 4. This is because the size of input image patch increases quickly from PedConv 1 to PedConv 4. The scaling factor between the input image patch of each detector is 1.2. The model window size of PedConv 4 is 220 × 110, which is about 1.7 times as big as that of PedConv 1. Therefore, the shared features learned from PedConv 1 do not perform so well in PedConv 4. As shown in Fig. 8, the bigger the model window is, the higher the validation error is. To ensure that our method maintains the detection performance, we adopt at most four detectors with model windows of different sizes.

5.3. Results

During testing phase, the images are sub-sampled. When using only one ConvNet detector, that is, the baseline PedConv 1, we use a scaling stride of 1.20 between each scale. The scaling stride increases by 1.2 as we add one more detector. Fig. 9 presents the detection results on INRIA dataset and Caltech-USA dataset, respectively. The results are measured in log-average miss-rate (MR, lower is better). The numbers 1, 2, 3, and 4 denote the number of detectors used during detection process. It can be seen that the MR increases a little every time we add one more detector. However, the detection performance maintains relatively stable. On the INRIA dataset, the MR only increases by 0.86% when all the four detectors are used. On the more challenging Caltech-USA dataset, the MR increases by 1.04% when all the four detectors are used.

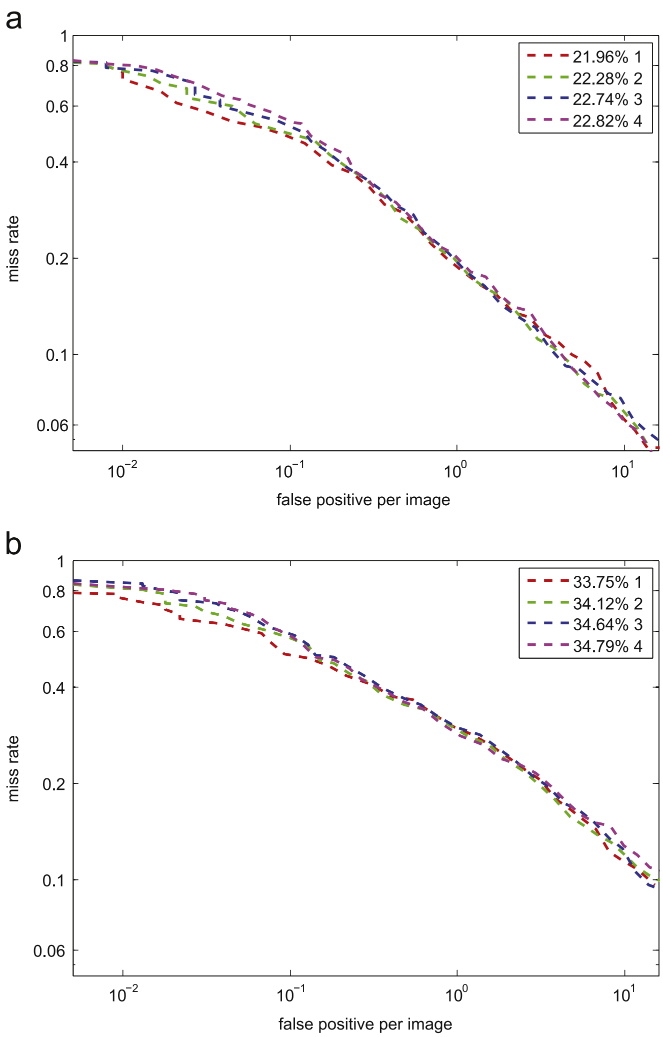


Fig. 9. Pedestrian detection results on: (a) the INRIA dataset, and (b) the Caltech-USA dataset. The detection performance maintains relatively stable when using four detectors by sharing features.

Table 1
Time spent on processing input images of three different sizes. The column denoted by 'Single' means we use PedConv 1 only. Columns denoted by 'Multi-scale (∗)' mean we use multiple models by sharing features.

Input image	Time (s)			
	Single	Multi-scale (2)	Multi-scale (3)	Multi-scale (4)
320 × 240	0.044	0.028	0.026	0.021
640 × 480	0.190	0.128	0.112	0.104
960 × 720	0.456	0.320	0.272	0.261

We use the time spent on processing one input image to evaluate the efficiency of our method. Table 1 presents the results. In our experiment, we use one single GTX 750 GPU which has only 2GB memory. As shown in Table 1, we test on images of three different sizes. The time decreases gradually as we use more detectors. Column denoted by 'Multi-scale (2)' means we use PedConv1 and PedConv 2 by sharing features. Column denoted by 'Multi-scale (3)' means we use PedConv1, PedConv 2, and PedConv 3 by sharing features. Column denoted by 'Multi-scale (4)' means we use all the four detectors by sharing features. Compared to one single detector, four detectors can decrease the processing time by approximately 50%. Take image of size 640 × 320 as an example, we can deal with almost 10 frames per second. Simultaneously, our method only results in a very small decline in detection performance.

6. Conclusion

We have demonstrated that our method can improve the efficiency of ConvNets for pedestrian detection. The improvement comes from two aspects. (1) We share features across a group of ConvNet based pedestrian detectors that have model windows of different sizes. (2) We simultaneously detect pedestrians of several scales on one single layer of an image pyramid. Compared to traditional scanning method, the proposed method can reduce the number of layers that are needed for extracting features. To make sure that the detection performance stays stable, we used at most four ConvNet detectors in our experiment. With four detectors, our method can decrease the detection time by almost 50%. Our future work will focus on further improving the detection efficiency of ConvNets for pedestrian detection.

Acknowledgement

This work was supported in part by the National Basic Research Program of China (973 Program) (Grant no. 2014CB340400), the National Natural Science Foundation of China (Grant nos. 61172121, 61271412, 61472274, 61222109, and 61503274), the Key Research Program of the Chinese Academy of Sciences (Grant no. KGZD-EW-T03), and Excellent Young Scholar of the Tianjin University of Technology and Education (Grant no. RC14-46).

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

References

[1] P. Agrawal, R. Girshick, J. Malik, Analyzing the performance of multilayer neural networks for object recognition, in: Proceedings of European Conference on Computer Vision, 2014, pp. 329–344.

[2] R. Benenson, M. Omran, J. Hosang, B. Schiele, Ten years of pedestrian detection, what have we learned? in: European Conference on Computer Vision, CVRSUAD workshop, 2014.

[3] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.

[4] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state-of-the-art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2012) 743–761.

[5] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.

[6] P. Fischer, A. Dosovitskiy, T. Brox, Descriptor matching with convolutional neural networks: a comparison to sift, arXiv:1405.5769, 2014.

[7] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[8] A. Giusti, D. Ciresan, J. Masci, L. Gambardella, J. Schmidhuber, Fast image scanning with deep max-pooling convolutional neural networks, in: Proceedings of IEEE International Conference on Image Processing, 2013, pp. 4034–4038.

[9] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Proceedings of European Conference on Computer Vision, 2014, pp 346–361.

[10] J. Hosang, M. Omran, R. Benenson, B. Schiele, Taking a deeper look at pedestrians, arXiv:1501.05790, 2015.

[11] S. Jeonga, Y. Parkb, R. Mallipeddia, J. Tanic, M. Leea, Goal-oriented behavior sequence generation based on semantic commands using multiple timescales recurrent neural network with initial state correction, Neurocomputing 129 (4) (2014) 67–77.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia, 2014, pp. 675–678.

[13] X. Jiang, Y. Pang, J. Pan, X. Li, Flexible sliding windows with adaptive pixel strides, Signal Process. 110 (2015) 37–45.

[14] A. Krizhevsky, Learning multiple layers of features from tiny images, Technical Report, University of Toronto, 2009.

[15] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, Proc. Adv. Neural Inf. Process. Syst. (2012) 1097–1105.

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, F. Li, Large-scale video classification with convolutional neural networks, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[17] M. Liu, S. Li, S. Shan, X. Chen, AU-inspired deep networks for facial expression feature learning, Neurocomputing 159 (2) (2015) 126–136.

[18] P. Luo, Y. Tian, X. Wang, X. Tang, Switchable deep network for pedestrian detection, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2014, pp. 899–906.

[19] G. Montaze, D. Giveki, An improved radial basis function neural network for object image retrieval, Neurocomputing 168 (11) (2015) 221–233.

[20] W. Ouyang, P. Luo, X. Zeng, S. Qiu, Y. Tian, H. Li, S. Yang, Z. Wang, Y. Xiong, C. Qian, Z. Zhu, R. Wang, C. Loy, X. Wang, X. Tang, DeepID-Net: multi-stage and deformable deep convolutional neural networks for object detection, arXiv:1409.3505, 2014.

[21] W. Ouyang, X. Wang, A discriminative deep model for pedestrian detection with occlusion handling, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 3258–3265.

[22] W. Ouyang, X. Wang, Joint deep learning for pedestrian detection, in: Proceedings of IEEE International Conference on Computer Vision, 2013, pp. 2056–2063.

[23] W. Ouyang, X. Wang, Single-pedestrian detection aided by multi-pedestrian detection, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2013, pp. 3198–3205.

[24] Y. Pang, K. Zhang, Y. Yuan, K. Wang, Distributed object detection with linear SVMs, IEEE Trans. Cybern. 44 (11) (2014) 2122–2133.

[25] A. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: an astounding baseline for recognition, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2014, pp. 512–519.

[26] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2013, pp. 3626–3633.

[27] Y. Pang, X. Jiang, X. Li, J. Pan, Efficient object detection by prediction in 3D space, Signal Process. 112 (2015) 64–73.

[28] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, Proc. Adv. Neural Inf. Process. Syst. (2014) 568–576.

[29] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556, 2014.

[30] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, .arXiv:1312.6229v4, 2013.

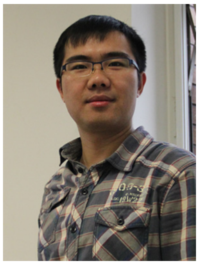
[31] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.

[32] S. Walk, N. Majer, K. Schindler, B. Schiele, New features and insights for pedestrian detection, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2010, pp. 1030–1037.

[33] C. Zitnick, P. Dollár, Edge boxes: locating object proposals from edges. in: Proceedings of European Conference on Computer Vision, 2014, pp. 391–405.

[34] M. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proceedings of European Conference on Computer Vision, 2014, pp. 818–833.

[35] X. Zeng, W. Ouyang, X. Wang, Multi-stage contextual deep learning for pedestrian detection, in: Proceedings of IEEE International Conference on Computer Vision, 2013, pp. 121–128.



Xiaoheng Jiang received the B.S. degree and M.S. degree in electronic engineering from the Tianjin University, China, in 2010 and 2013, respectively. He is currently a Ph.D. candidate in the Tianjin University. His research interests include deep learning, object detection and image analysis.

Yanwei Pang received the Ph.D. degree in electronic engineering from the University of Science and Technology of China, Hefei, China, in 2004. He is currently a Professor with the School of Electronic Information Engineering, Tianjin University, Tianjin, China. His current research interests include object detection and recognition, and image processing. He has published more than 90 scientific papers. His current research interests include object detection and recognition, and image processing.

Xuelong Li is with the 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, China.

Jing Pan received her B.S. degree in Mechanical Engineering from the North China Institute of Technology (now North University of China), Taiyuan, China, in 2002, and her M.S. degree in Precision Instrument and Mechanism from the University of Science and Technology of China, Hefei, China, in 2007. She is currently a Lecturer with the School of Electronic Engineering, Tianjin University of Technology and Education, Tianjin, China. Meanwhile, she is pursuing her Ph.D. degree in the Tianjin University, China. Her research interests include computer vision and pattern recognition.

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