Late Fusion of Multiple Convolutional Layers for Pedestrian Detection

Ujjwal
Vedecom and INRIA
Sophia Antipolis, France
ujjwal@vedecom.fr

Aziz Dziri, Bertrand Leroy
Vedecom
Versailles, France
aziz.dziri@vedecom.fr, bertrand.leroy@vedecom.fr

Francois Bremond
INRIA
Sophia Antipolis, France
francois.bremond@inria.fr

Abstract

We propose a system design for pedestrian detection by leveraging the power of multiple convolutional layers explicitly. We quantify the effect of different convolutional layers on the detection of pedestrians of varying scales and occlusion level. We show that earlier convolutional layers are better at handling small-scale and partially occluded pedestrians. We take cue from these conclusions and propose a pedestrian detection system design based on Faster-RCNN which leverages multiple convolutional layers by late fusion. In our design, we introduce height-awareness in the loss function to make the network emphasize on pedestrian heights which are misclassified during the training process. The proposed system design achieves a log-average miss-rate of 9.25% *on the caltech-reasonable dataset. This is within* 1.5% *of the current state-of-art approach, while being a more compact system.*

1. Introduction

One of the prominent problems in pedestrian detection is handling scale and occlusion. These problems are quite well aligned with the recent interests in autonomous vehicles. Successful detection of far-scale pedestrians can assist the vehicle in making safety maneuvers well ahead in time, thereby promoting a safer traffic environment. The same is true for surveillance systems in high security environments like airports and ports [4].

The objective of this paper is to design a pedestrian detection system for handling the aforementioned problems. We consider Faster-RCNN based detection framework as a basis for this design. This is motivated by the fact that Faster-RCNN framework represents the crucial components of a broad family of object detection techniques including SSD [18], RPN-BF [16] and SDS-RCNN [5], which are used for pedestrian detection. For our design, show that lower layers detect small-scale and occluded pedestrians better than later layers; which are better at detecting large-scale and unoccluded pedestrians. We take cue from these conclusions and present a system design which performs at par with the state-of-art pedestrian detection systems.

2. Related work

Most of the recent approaches to pedestrian detection are based on deep learning approaches. This usually involves extracting features from a Convolutional Neural Network (CNN) and subsequently classifying them using fully-connected layers [18, 11, 7] or other classifiers like boosted forests [16] and cascade of deep networks [3]. Localization of objects of interest is done by using sliding windows called anchor boxes [11, 18, 7] or a subwindow search [8]. There have been some non-deep learning approaches to pedestrian [14, 19, 23]. However, they have been outperformed by deep learning approaches.

Many deep learning approaches incorporate additional constraints to facilitate improved learning such as parts-based detection [21, 13], improved classification between pedestrian and background samples using transfer learning from segmentation datasets [22, 2].

Utilizing multiple layers of CNNs has also evoked some recent interest. SSD[18] is an early example of late fusion of detections from multiple layers. In [17], the authors propose foveal regions through skip connections and call their network as a multipath network. It does utilize multiple convolutional layers simultaneously for creating these foveal regions. They utilize segmentation proposals thereby limiting their usability in datasets without segmentation information. In [20], the authors utilize features of multiple convolutional layers and use an unsupervised scheme to pre-train their filters. However, their evaluation protocol has been slightly different [20], and we do not get a good insight into the role of individual layers on pedestrian scale and occlusion. In this work we attempt to understand the impact of different convolutional layers on pedestrian scale and occlusion. Compactness and deployability is another limitation of most recent works on pedestrian detection [2, 16, 5]. The systems become bulky and it is often hard to extend them to novel architectures (e.g- porting SSD to ResNet-101). In our work the system is more compact and offers crucial insights about the role of individual layers, thereby facilitating extension and generalization to other architectures. Following our study, we design a detection system utilizing multiple layers explicitly for pedestrian detection.

3. Datasets and Evaluation Metrics

3.1. Datasets

In this work we utilize the Caltech pedestrian dataset [1] for benchmarking. Availability of large number of benchmarks [16, 5, 14] make it a vital dataset for studying our work and its impact.

The resolution of Caltech dataset images is 640×480 and has been captured from a moving vehicle without any corrections for vehicle pitching [1]. The original annotations in Caltech suffered from alignment problems [15]. We utilize the improved annotations [15] in our work for benchmarking. Following [15] the training images taken at interval of 3 frames (caltech10x-train) are used for training. The testing images taken at interval of 30 frames (caltech1x-test) are used for testing. We have also evaluated our method on the reasonable subset (height $>$ 50px and occlusion < 0.35).

Evaluation on the reasonable subset involves evaluating on the reasonable annotations from the caltech1x-test set. New annotations of [15] are available only for these subsets (caltech10x and caltech1x).

3.2. Evaluation Metrics

We use log-average miss rate as proposed in [1] for the evaluation of our proposed approach. This average miss-rate is computed for a range of *false positives per image* (FPPI) in $[10^{-2}, 10^{0}]$.

4. Proposed Approach

4.1. Layer-wise analysis of CNN layers' effect on scale and occlusion

As features pass between convolutional layers separated by pooling layers, they undergo feature and scale transformation. Intuitively, it suggests that small-scale and partially occluded pedestrians can be captured well by the lower CNN layers. To look into this effect quantitatively, we design a system as exhibited in Figure 1. For our study, we use the data flow depicted in figure 1 using dashed green lines, i.e, no concatenation is performed. It uses VGG16 [10] as a base network which is also used in Faster-RCNN [11]. We model a detection framework after Faster-RCNN, by extracting features from a subset of convolutional layers. VGG16 is divided into convolutional blocks. Within each block, feature maps are of the same shape and they are transformed into half their size by the pooling layer between consecutive blocks [10]. We extract features (f_i^{conv}) from the last convolutional layer within each block (thus $i \in \{1, 2..5\}$. Each f_i^{conv} feeds into a separate region proposal network (RPN) [11]. A RPN passes the feature map through one or more convolutional layers resulting in a feature map (f_{map}^{RPN}). It then slides a set of template bounding boxes (*anchors*) over f_{map}^{RPN}. An anchor is a template bounding box which is translated through a feature map. It is centered at each pixel and for the resulting area of the feature map it determines if that area contains a region of interest [11]. Anchors' intersection over union (IoU) with groundtruth bounding boxes is used to determine the presence of positive proposals during training (IoU$> 0.7 \implies$ +ve and IoU$< 0.3 \implies$ -ve, $0.3 <$IoU< 0.7 don't contribute to RPN training) [11]. Anchors are created for varying scale and aspect ratios. Due to the upright nature of pedestrians, pedestrian specific systems [16, 5] consider anchors encompassing several scales but limited to a constant aspect ratio of 0.41 (*mode of aspect ratios of pedestrians in caltech dataset*). Using two sibling fully connected layers, RPN performs two tasks on these anchors – a) proposal classification and b) proposal bounding box regression. In Faster-RCNN, positively classified proposals are further refined by a set of two sibling fully-connected layers for final object-level classification and regression. In a related work [16], the authors show an improved performance using boosted forests. We use gradient boosted trees as classifiers, to do the final object level classification. At the end we use soft non-maximal suppression [6] for getting final detections

In our experiments we use one convolutional layer (*kernel size*: 3×3, *stride*: 1, *padding*: 1, *num-filters*: 512) in each RPN. More convolutional layers for early layer feature maps may be beneficial but they require much GPU memory. Our base VGG16 network is pre-trained on the imagenet dataset. We train the design of figure 1, in two stages. In the first stage, we train the RPNs, using stochastic gradient descent with the same loss function (eqn 1) as proposed

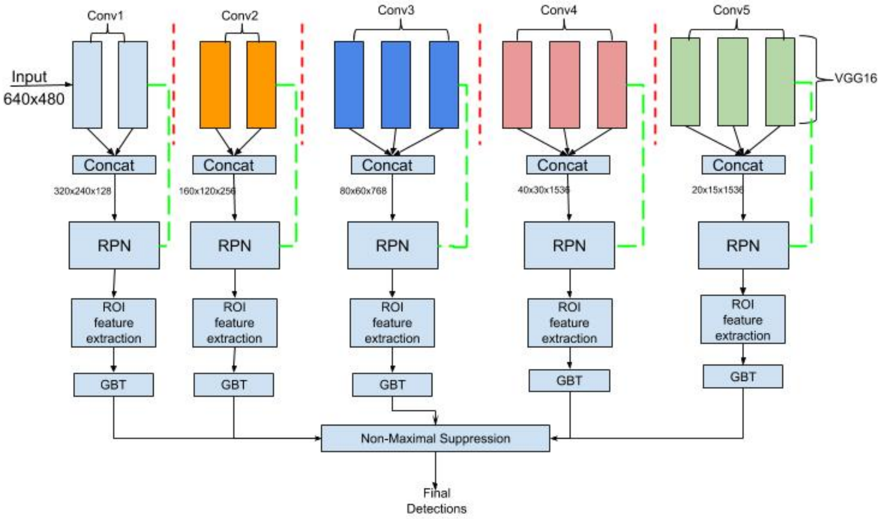

Figure 1. Block Diagram of the proposed pedestrian detection system. The data flow for the study in section 4.1, is indicated using dashed green lines. We later improve upon this and the final system's data flow is shown in solid black arrows. The red dashed lines in the diagram refer to the location of pooling layers in VGG16 [10], where the feature map changes size.

in [11].

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \frac{\lambda}{N_{reg}}\sum_i L_{reg}(t_i,t_i^*)p_i^* \quad (1)$$

In equation 1, quantities with a *star* (p_i^*, t_i^*) refer to groundtruth and those without it refer to the anchor which overlapped them with an IoU>0.7. $\{p_i\}$ is the label of i^{th} anchor (+ve or -ve), $\{t_i\}$ is the vector of bounding box coordinates of i^{th} anchor, L_{cls} is the cross-entropy loss, L_{reg} is the smoothed-L1 loss, λ is a scalar constant (set to 1.0 in our implementation). N_{cls} is minibatch size and N_{reg} is total number of anchors. Following the RPN training, we select the top 1500 ROIs based on RPN scores and extract RPN features from them. These serve as input to gradient boosted trees (GBT). We train the GBTs using the XGBoost framework. We set the maximum depth of each tree as 6, and the maximum number of trees as 1024. GBTs are used only for classification. Coordinates predicted by the regression layer of RPN are not further regressed by GBTs.

To evaluate the impact of the convolutional layer connected to i^{th} RPN (RPN_i), we detach all the other RPNs from the non-maximal suppression. We use the caltech10x training set [15] for training and caltech1x-testing set [15]. For occlusion studies we also use the complete caltech test dataset for testing using old annotations due to wider range of occlusion levels present in them.

4.1.1 Effect of CNN layers on scale and occlusion based detection

Table 1 shows the log-average miss-rate for different pedestrian heights in the caltech-reasonable test-set. The column titled "fused", refers to the configuration with all RPNs contributing to NMS. For larger heights, better performance is achieved by the later layers, while the opposite is true for smaller heights. Small-scale pedestrians are mainly discriminated by their contours and other low-level features. Near-scale pedestrians have greater amount of visual detail, which varies with context such as clothing. Thus, their detection requires more semantic features which are captured by higher CNN layers. Moreover, due to a sequence of pooling layers, feature resolution decreases with CNN layer depth. For small-scale pedestrians this leads to their features reduced to sub-pixel accuracy. This is another factor behind a lower accuracy of detection of small-scale pedestrians in later CNN layers.

Table 2, shows the performance of different layers based on pedestrian occlusion. We test the system on the complete caltech dataset. To keep the study concise and tractable, we limit ourselves to pedestrians with a minimum height of 50 pixels. Extremely occluded pedestrians (*occlusion* $>$ *80%*) often require specialized approaches and we keep them out of our study. Table 2, points to the conclusion that lower CNN layers capture occlusion better than higher layers. This is expected as high-level of occlusions imply a less amount of visible pedestrian area, which is compromised by pooling layers. In the caltech dataset, high levels of occlusion are primarily observed in medium-to-small scale pedestrians [1]. We consider the trends of tables 1 and

Pedestrian height(in pixels)	**Layer 1**	**Layer 2**	**Layer 3**	**Layer 4**	**Layer 5**	**Fused**
> 80	4.83	4.97	3.88	2.17	**2.15**	3.27
$50 - 80$	**9.17**	9.54	9.40	10.48	10.68	10.43
All (height > 50)	12.83	13.12	11.34	11.2	**10.95**	10.16

Table 1. Log-Averaged miss rate for pedestrians of different heights by different layers in Caltech-reasonable (*test*) dataset.

Occlusion($\%$ occlusion by area)	**Layer 1**	**Layer 2**	**Layer 3**	**Layer 4**	**Layer 5**	**Fused**
0 (No occlusion)	10.6	10.4	9.7	9.68	**9.64**	9.79
$1 - 35$	**22.3**	24.75	25.2	25.5	26.1	25.3
$65 - 80$	**67.2**	69.9	71.2	73.5	76	75.4
All ($0 - 80\%$ occlusion)	**68.4**	70.1	72.7	75	77.2	76.3

Table 2. Log-Averaged miss rate for varying occlusion levels by different layers in Caltech-complete (test) dataset. For the Caltech-complete we have used the old annotations (pedestrian height > 50 pixels.)

2, fairly indicative and frame our design from these conclusions.

4.2. Pedestrian Detection System Design

Following the inferences from section 4.1, we design our pedestrian detection system to leverage multiple convolutional layers explicitly and simultanously. Our system design is similar to the one we used in our study (figure 1), with some modifications. The system level modifications include –

1. Concatenating all convolutional layers within a block before feeding the corresponding RPN.
2. Using a dense system of pedestrian specific anchors.
3. Use of a modified loss function for RPN.

4.2.1 Feature Map Concatenation

As shown in figure 1, we concatenate feature maps from convolutional layers in each block of VGG16. VGG16 feature maps in each block have the same dimensions. Thus they can be concatenated without any overhead of resizing. Due to hierarchical character of CNN features, this also enhances the feature diversity for the input to the RPN.

4.2.2 Anchor Design

Some preliminary bounds are available to determine the minimum detectable size of an object for an anchor of identical scale as the object [9]. However their assumption requires us to have cover all possible scales of objects which is impractical for large datasets like caltech. To mitigate the risk of choosing a set of fixed anchors, we consider a more dense set of anchors (*large range of scales with small step sizes*) as compared to [16] and [5]. Our anchor construction is detailed in table 3. All our anchors have a fixed aspect ratio of 0.41 (*width/height*). Anchors which fall outside a feature map perimeter are eliminated from computations. A dense set of anchors though increasing computational complexity, helps in improving detection of positive (*pedestrian*) region proposals.

Layer	**Scales**
Conv1	[12, 512, 32]
Conv2	[10, 256, 32]
Conv3	[8,128, 32]
Conv4	[8, 128, 16]
Conv5	[8, 128,16]

Table 3. Anchor scales chosen for different layers. The notation [A,B,C] in the second column refers to minimum scale as A, maximum scale as B with a step-size of C (all in pixels).

4.2.3 Loss Function

The loss function for RPNs in [11, 16], do not take into account the object dimensions. In our work we give weight to the observation that pedestrians can appear at a wide variety of scales on account of their varying distances from a camera. To this end, we modify the RPN loss function as follows–

$$L(\{p_i\},\{t_i\}) \triangleq \frac{1}{N_i}\sum_i [I_i^1\{\frac{L_{cls}(p_i,p_i^*)+\zeta L_{reg}(t_i,t_i^*)}{f(h_i)+\epsilon}\}+ \eta I_i^0 L_{cls}(p_i,p_i^*)] \quad (2)$$

In equation 2, I_i^1 is an indicator function which is 1 if the i^{th} anchor is a pedestrian candidate proposal and 0 otherwise. I_i^0 is an indicator function which is 1 if the i^{th} anchor is a non-pedestrian candidate proposal and 0 otherwise. L_{cls} is the log-loss over two classes (*pedestrian* vs. *non-pedestrian*). L_{reg} is the regression loss and it is clear from equation 2, that it is activated only for positive proposals. ζ and η are scalar constants. ζ intuitively denotes the balance between the classification loss and the regression loss. η denotes the balance between the true-positive(TP) classification and true-negative(TN) classification. In our experiments we set both ζ and η to 1. We normalize our loss function by the total number of anchors in the RPN (N_i). Other symbols (p_i,p_i^*,t_i,t_i^*) are the same as in equation 1. In our loss function $f(h_i)$ denotes the running accuracy of correct RPN classification for a pedestrian of height h_i. Running accuracy is a metric which keeps an updated track of the accuracy through the iterations of training. The running accuracy is computed as follows-

- **Step 1:** Get all unique heights $H \triangleq \{h_1, h_2, ..., h_N\}$ in the dataset.
- **Step 2:** Initialize a dictionary D with elements of H as keys. For each h_i in H, $D[h_i] \triangleq (CC_i, TC_i)$. Here, $(.,.)$ is the notation for a tuple of two numbers. CC_i, is the cumulative count (*across iterations*) of correct RPN classifications for a pedestrian of height h_i. TC_i is the cumulative count (*across iterations*) of the number of times, a positive proposal overlapping with a pedestrian of height h_i has passed through the proposal classification layer.
- **Step 3:** The running accuracy $f(h_i)$ is then defined as : $f(h_i) \triangleq \frac{CC_i}{TC_i}$

Running accuracy is just an online version of accuracy of prediction. The parameter ϵ is a small scalar constant to avoid infinity, when $f(h_i) = 0$. The incorporation of $f(h_i)$, makes the loss function in equation 2, scale aware. For pedestrians of certain heights, which are not getting detected by RPN, the loss function is penalized. For every unique height of pedestrians, $f(h_i)$ is maintained. During the training process, feature weights in the network are constantly updated. Some of these updates may improve detections of certain heights while deteriorating other pedestrian heights. Keeping a continuous track of accuracy during training helps in stablizing the network towards a balance of detection of all pedestrian heights. This is in stark contrast to [11, 16], where the network is not made to give any weightage to the pedestrian height.

For bounding box regression we use the same smoothed L1-loss and bounding box parametrization as adopted in [11].

5. Experiments and Results

5.1. Training

We train each RPN using the SGD optimizer for 30 epochs with a learning rate of 0.001 and a momentum of 0.99. The learning rate was decreased by a factor of 10 after every 10 epochs. A total of 4 Nvidia Titan X (Maxwell) GPUs; each with 12 GB of memory; were utilized during training. We use the caltech 10x training set and caltech 1x testing set in our experiments as in section 4.1. During the training process, the convolutional layers of the base network are also modified. Based on our experiments, we found that the RPNs should be trained starting from the lowest convolutional layers and down to the highest. Moreover a RPN should be allowed to modify only the convolutional layers, whose feature maps are concatenated to supply the input to the RPN. This helps in making the training process more stable.

5.2. Results

In table 4 we show the performance of the proposed system on two variations of the caltech 1x testing set. In the first row, we show the performance on the reasonable subset (*height* > 50 px and *occlusion* $< 35\%$). In the second row results are shown for all annotations in the caltech 1x testing set. Table 5 shows a comparison of our performance vis-à-vis other recent methods; revealing a comparable and competitive performance of our approach. Table 6 shows the height-wise performance of the proposed approach over the caltech-all test dataset.

6. Analysis

Our work demonstrates that combining multiple convolutional layers can improve pedestrian detection. Our approach to combination is late fusion, where all convolutional layers are trained to detect all pedestrian heights. This is a slight detour from our study in section 4.1, which reveals differing affinities of different layers for scale and occlusion. It is hence, beneficial to explore approaches to make individual layers more scale and occlusion specific, where a layer is specialized to detect pedestrians in a specific scale or occlusion range. We consider this a future objective of our work. We have also explored the use of running accuracy metric in the RPN loss function. While it has helped in faster training, its precise impact needs to be verified by ablation studies. Inclusion of $f(h_i)$ directly in denominator can give big jumps to the loss function thereby reducing the stability of training. Exploring other weightage functions is also an important part of our future work.

7. Conclusions

Our work highlights the role of multiple convolutional layers in detecting pedestrians. We show that different scale and occlusion levels of pedestrians are captured with different accuracies by different layers. Our system built from these cues performs competitively against other popular approaches. We have also laid down important directions for future work on this approach. We believe that with further

Subset	**Log-Average Miss Rate (Testing)**	**Testing Set**	**Training Set**
Caltech Reasonable	9.25%	Caltech-reasonable (1x-Test)	Caltech-reasonable (10x-Train)
Caltech-All	46.2%	Caltech-all (1x-Test)	Caltech-all (10x-Train)

Table 4. Log-averaged miss rate over different subsets of caltech[12]. The testing and training subsets are shown in the 3^{rd} and 4^{th} columns. New caltech annotations of [12] are used for training and testing.

Method	**Caltech-Reasonable(1x) Test Set**
Faster-RCNN[11]	15%
RPN-BF[16]	9.58%
RPN-BF (Using only RPN)[16]	14.8%
SDS-RPN[5]	9.63%
SDS-RCNN[5]	7.6%
MS-CNN[7]	9.95%
FDNN[2]	8.65%
Ours	**9.25%**

Table 5. Comparison with other works with performance on Caltech-reasonable(test set) of caltech1x

Heights	**Log-Average Miss Rate (Caltech-All Testing)**
>80px	3.25%
50-80px	12.66%
<50px	64.20%

Table 6. Miss-Rates for different pedestrian height ranges in the caltech-all testing (1x-test) set. This includes all occlusion levels. New annotations from [12] are used for training and testing.

work, this can be a potential direction for designing compact systems for high quality pedestrian detection.

References

[1] P. e. Dollar. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 2012.

[2] X. e. Du. Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection. In *WACV*. IEEE, 2017.

[3] A. et.al. Pedestrian detection with a large-field-of-view deep network. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015.

[4] B. et.al. Learning object motion patterns for anomaly detection and improved object detection. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008.

[5] B. et.al. Illuminating pedestrians via simultaneous detection & segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017.

[6] B. et.al. Improving object detection with one line of code. *arXiv preprint arXiv:1704.04503*, 2017.

[7] C. et.al. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*. Springer, 2016.

[8] E. et.al. Scalable object detection using deep neural networks. In *CVPR*, 2014.

[9] E. et.al. Improving small object proposals for company logo detection. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. ACM, 2017.

[10] S. et.al. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[11] S. R. et.al. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[12] S. Z. et.al. How far are we from solving pedestrian detection? In *CVPR*, 2016.

[13] W. et.al. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Computer Vision, 2005. ICCV 2005*. IEEE, 2005.

[14] Z. et.al. Filtered channel features for pedestrian detection. In *CVPR*, 2015.

[15] Z. et.al. How far are we from solving pedestrian detection? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[16] Z. et.al. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*. Springer, 2016.

[17] Z. et.al. A multipath network for object detection. *arXiv preprint arXiv:1604.02135*, 2016.

[18] W. e. Liu. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 2016.

[19] W. e. Nam. Local decorrelation for improved pedestrian detection. In *NIPS*, 2014.

[20] P. e. Sermanet. Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*. IEEE, 2013.

[21] Y. e. Tian. Deep learning strong parts for pedestrian detection. In *CVPR*, 2015.

[22] Y. e. Tian. Pedestrian detection aided by deep learning semantic tasks. In *CVPR*, 2015.

[23] S. Zhang, C. Bauckhage, and A. B. Cremers. Informed haar-like features improve pedestrian detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 947–954. IEEE, 2014.