Learning Efficient Single-stage Pedestrian Detectors by Asymptotic Localization Fitting

Wei Liu[1,3*], Shengcai Liao[1,2**], Weidong Hu[3], Xuezhi Liang[1,2], Xiao Chen[3]

[1] Center for Biometrics and Security Research & National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] National University of Defense Technology, Changsha, China
{liuwei16, wdhu, chenxiao15}@nudt.edu.cn, scliao@nlpr.ia.ac.cn, xzliang@cbsr.ia.ac.cn

Abstract. Though Faster R-CNN based two-stage detectors have witnessed significant boost in pedestrian detection accuracy, it is still slow for practical applications. One solution is to simplify this working flow as a single-stage detector. However, current single-stage detectors (e.g. SSD) have not presented competitive accuracy on common pedestrian detection benchmarks. This paper is towards a successful pedestrian detector enjoying the speed of SSD while maintaining the accuracy of Faster R-CNN. Specifically, a structurally simple but effective module called *Asymptotic Localization Fitting* (ALF) is proposed, which stacks a series of predictors to directly evolve the default anchor boxes of SSD step by step into improving detection results. As a result, during training the latter predictors enjoy more and better-quality positive samples, meanwhile harder negatives could be mined with increasing IoU thresholds. On top of this, an efficient single-stage pedestrian detection architecture (denoted as ALFNet) is designed, achieving state-of-the-art performance on CityPersons and Caltech, two of the largest pedestrian detection benchmarks, and hence resulting in an attractive pedestrian detector in both accuracy and speed. Code is available at https://github.com/VideoObjectSearch/ALFNet.

Keywords: Pedestrian Detection · Convolutional Neural Networks · Asymptotic Localization Fitting

1 Introduction

Pedestrian detection is a key problem in a number of real-world applications including auto-driving systems and surveillance systems, and is required to have both high accuracy and real-time speed. Traditionally, scanning an image in a sliding-window paradigm is a common practice for object detection. In this paradigm, designing hand-crafted features [11,2,10,29] is of critical importance for state-of-the-art performance, which still remains as a difficult task.

* Wei Liu finished his part of work during his visit in CASIA.
** Shengcai Liao is the corresponding author.

Beyond early studies focusing on hand-craft features, RCNN [17] firstly introduced CNN into object detection. Following RCNN, Faster-RCNN [32] proposed Region Proposal Network (RPN) to generate proposals in a unified framework. Beyond its success on generic object detection, numerous adapted Faster-RCNN detectors were proposed and demonstrated better accuracy for pedestrian detection [42,44]. However, when the processing speed is considered, Faster-RCNN is still unsatisfactory because it requires two-stage processing, namely proposal generation and classification of ROIpooling features. Alternatively, as a representative one-stage detector, Single Shot MultiBox Detector (SSD) [27] discards the second stage of Faster-RCNN [32] and directly regresses the default anchors into detection boxes. Though faster, SSD [27] has not presented competitive results on common pedestrian detection benchmarks (e.g. CityPersons [44] and Caltech [12]). It motivates us to think what the key is in Faster R-CNN and whether this key could be transfered to SSD. Since both SSD and Faster R-CNN have default anchor boxes, we guess that the key is the two-step prediction of the *default anchor boxes*, with RPN one step, and prediction of ROIs another step, but not the ROI-pooling module. Recently, Cascade R-CNN [6] has proved that Faster R-CNN can be further improved by applying multi-step ROI-pooling and prediction after RPN. Besides, another recent work called RefineDet [45] suggests that ROI-pooling can be replaced by a convolutional transfer connection block after RPN. Therefore, it seems possible that the default anchors in SSD could be directly processed in multi-steps for an even simpler solution, with neither RPN nor ROI-pooling.

Another problem for SSD based pedestrian detection is caused by using a single IoU threshold for training. On one hand, a lower IoU threshold (e.g. 0.5) is helpful to define adequate number of positive samples, especially when there are limited pedestrian instances in the training data. For example, as depicted in Fig.1 (a), the augmented training data [42] on Caltech has 42782 images, among which about 80% images have no pedestrian instances, while the remains have only 1.4 pedestrian instances per image. However, a single lower IoU threshold during training will result in many "close but not correct" false positives during inference, as demonstrated in Cascade R-CNN [6]. On the other hand, a higher IoU threshold (e.g. 0.7) during training is helpful to reject close false positives during inference, but there are much less matched positives under a higher IoU threshold, as pointed out by Cascade R-CNN and also depicted in Fig.1 (b). This positive-negative definition dilemma makes it hard to train a high-quality SSD, yet this problem is alleviated by the two-step prediction in Faster R-CNN.

The above analyses motivate us to train the SSD in multi-steps with improving localization and increasing IoU thresholds. Consequently, in this paper a simple but effective module called Asymptotic Localization Fitting (ALF) is proposed. It directly starts from the default anchors in SSD, and convolutionally evolves all anchor boxes step by step, pushing more anchor boxes closer to groundtruth boxes. On top of this, a novel pedestrian detection architecture is constructed, denoted as Asymptotic Localization Fitting Network

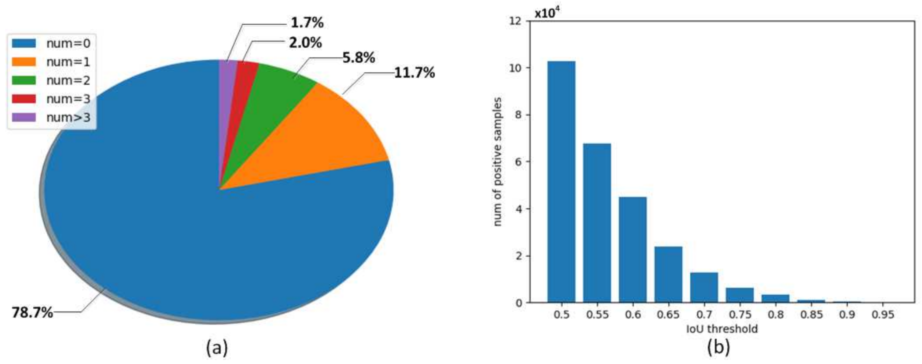


Fig. 1. (a) Percentage of images with different number of pedestrian instances on the Caltech training dataset newly annotated by [43]. (b) Number of positive anchors w.r.t. different IoU threshold. Each bar represents the number of default anchors matched with any ground truth higher than the corresponding IoU threshold.

(ALFNet). ALFNet significantly improves the pedestrian detection accuracy while maintaining the efficiency of single-stage detectors. Extensive experiments and analysis on two large-scale pedestrian detection datasets demonstrate the effectiveness of the proposed method independent of the backbone network.

To sum up, the main contributions of this work lie in: (1) a module called ALF is proposed, using multi-step prediction for asymptotic localization to overcome the limitations of single-stage detectors in pedestrian detection; (2) the proposed method achieves new state-of-the-art results on two of the largest pedestrian benchmarks (i.e., CityPerson [44], Caltech [12]).

2 Related work

Generally, CNN-based generic object detection can be roughly classified into two categories. The first type is named as two-stage methods [17,16,32,8], which first generates plausible region proposals, then refines them by another sub-network. However, its speed is limited by repeated CNN feature extraction and evaluation. Recently, in the two-satage framework, numerous methods have tried to improve the detection performance by focusing on network architecture [8,22,23,25], training strategy [34,39], auxiliary context mining [1,15,35], and so on, while the heavy computational burden is still an unavoidable problem. The second type [27,30,31], which is called single-stage methods, aims at speeding up detection by removing the region proposal generation stage. These single-stage detector directly regress pre-defined anchors and thus are more computationally efficient, but yield less satisfactory results than two-stage methods. Recently, some of these methods [14,33] pay attention to enhancing the feature representation of CNN, and some others [21,26] target at the positive-negative imbalance problem via novel classification strategies. However, less work has been done for pedestrian detection in the single-stage framework.

In terms of pedestrian detection, driven by the success of RCNN [17], a series of pedestrian detectors are proposed in the two-stage framework. Hosang et al. [19] firstly utilizes the SCF detector [2] to generate proposals which are then fed into a RCNN-style network. In TA-CNN [38], the ACF detector [10] is employed for proposal generation, then pedestrian detection is jointly optimized with an auxiliary semantic task. DeepParts [37] uses the LDCF detector [29] to generate proposals and then trains an ensemble of CNN for detecting different parts. Different from the above methods with resort to traditional detectors for proposal generation, RPN+BF [42] adapts the original RPN in Faster-RCNN [32] to generate proposals, then learns boosted forest classifiers on top of these proposals. Towards the multi-scale detection problem, MS-CNN [4] exploits multi-layers of a base network to generate proposals, followed by a detection network aided by context reasoning. SA-FastRCNN [24] jointly trains two networks to detect pedestrians of large scales and small scales respectively, based on the proposals generated from ACF detector [10]. Brazil et al. [3], Du et al. [13] and Mao et al. [28] further improve the detection performance by combining semantic information. Recently, Wang et al. [40] designs a novel regression loss for crowded pedestrian detection based on Faster-RCNN [32], achieving state-of-the-art results on CityPersons [44] and Caltech [12] benchmark. However, less attention is paid to the speed than the accuracy.

Most recently, Cascade R-CNN [6] proposes to train a sequence of detectors step-by-step via the proposals generated by RPN. The proposed method shares the similar idea of multi-step refinement to Cascade R-CNN. However, the differences lie in two aspects. Firstly, Cascade R-CNN is towards a better detector based on the Faster R-CNN framework, but we try to answer what the key in Faster R-CNN is and whether this key could be used to enhance SSD for speed and accuracy. The key we get is the multi-step prediction, with RPN one step, and prediction of ROIs another step. Given this finding, the default anchors in SSD could be processed in multi-steps, in fully convolutional way without ROI pooling. Secondly, in the proposed method, all default anchors are convolutionally processed in multi-steps, without re-sampling or iterative ROI pooling. In contrast, the Cascade R-CNN converts the detector part of the Faster R-CNN into multi-steps, which unavoidably requires RPN, and iteratively applying anchor selection and individual ROI pooling within that framework.

Another close related work to ours is the RefineDet [45] proposed for generic object detection. It contains two inter-connected modules, with the former one filtering out negative anchors by objectness scores and the latter one refining the anchors from the first module. A transfer connection block is further designed to transfer the features between these two modules. The proposed method differs from RefineDet [45] mainly in two folds. Firstly, we stack the detection module on the backbone feature maps without the transfer connection block, thus is simpler and faster. Secondly, all default anchors are equally processed in multi-steps without filtering. We consider that scores from the first step are not confident enough for decisions, and the filtered ”negative” anchor boxes may contain hard positives that may still have chances to be corrected in latter steps.

3 Approach

3.1 Preliminary

Our method is built on top of the single-stage detection framework, here we give a brief review of this type of methods.

In single-stage detectors, multiple feature maps with different resolutions are extracted from a backbone network (e.g. VGG [36], ResNet [18]), these multi-scale feature maps can be defined as follows:

$$\varPhi_n = f_n(\varPhi_{n-1}) = f_n(f_{n-1}(...f_1(I))), \tag{1}$$

where I represents the input image, $f_n(.)$ is an existing layer from a base network or an added feature extraction layer, and $\varPhi_n$ is the generated feature maps from the nth layer. These feature maps decrease in size progressively thus multi-scale object detection is feasible of different resolutions. On top of these multi-scale feature maps, detection can be formulated as:

$$Dets = F(p_n(\varPhi_n, \mathcal{B}_n), p_{n-1}(\varPhi_{n-1}, \mathcal{B}_{n-1}), ..., p_{n-k}(\varPhi_{n-k}, \mathcal{B}_{n-k})), n > k > 0, \tag{2}$$

$$p_n(\varPhi_n, \mathcal{B}_n) = \{cls_n(\varPhi_n, \mathcal{B}_n), regr_n(\varPhi_n, \mathcal{B}_n)\}, \tag{3}$$

where $\mathcal{B}_n$ is the anchor boxes pre-defined in the nth layer's feature map cells, $p_n(.)$ is typically a convolutional predictor that translates the nth feature maps $\varPhi_n$ into detection results. Generally, $p_n(.)$ contains two elements, $cls_n(.)$ which predicts the classification scores, and $regr_n(.)$ which predicts the scaling and offsets of the default anchor boxes associated with the nth layer and finally gets the regressed boxes. $F(.)$ is the function to gather all regressed boxes from all layers and output final detection results. For more details please refer to [27].

We can find that Eq. (2) plays the same role as RPN in Faster-RCNN, except that RPN applies the convolutional predictor $p_n(.)$ on the feature maps of the last layer for anchors of all scales (denoted as $\mathcal{B}$), which can be formulated as:

$$Proposals = p_n(\varPhi_n, \mathcal{B}), n > 0 \tag{4}$$

In two-stage methods, the region proposals from Eq. (4) is further processed by the ROI-pooling and then fed into another detection sub-network for classification and regression, thus is more accurate but less computationally efficient than single-stage methods.

3.2 Asymptotic Localization Fitting

From the above analysis, it can be seen that the single-stage methods are suboptimal primarily because it is difficult to ask a single predictor $p_n(.)$ to perform perfectly on the default anchor boxes uniformly paved on the feature maps. We argue that a reasonable solution is to stack a series of predictors $p_n^t(.)$

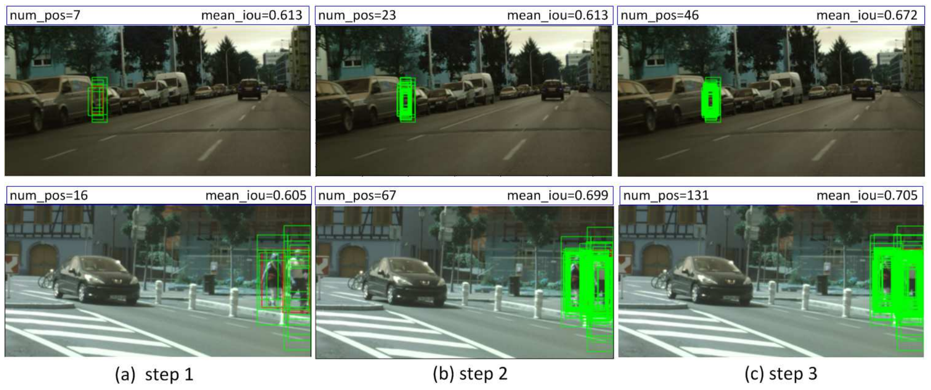


Fig. 2. Two examples from the CityPersons [44] training data. Green and red rectangles are anchor boxes and groundtruth boxes, respectively. Values on the upper left of the image represent the number of anchor boxes matched with the groundtruth under the IoU threshold of 0.5, and values on the upper right of the image denote the mean value of overlaps with the groundtruth from all matched anchor boxes.

applied on coarse-to-fine anchor boxes $\mathcal{B}_n^t$, where t indicates the t_{th} step. In this case, Eq. 3 can be re-formulated as:

$$p_n(\Phi_n, \mathcal{B}_n^0) = p_n^T(p_n^{T-1}(...(p_n^1(\Phi_n, \mathcal{B}_n^0))), \tag{5}$$

$$\mathcal{B}_n^t = regr_n^t(\Phi_n, \mathcal{B}_n^{t-1}), \tag{6}$$

where T is the number of total steps and $\mathcal{B}_n^0$ denotes the default anchor boxes paved on the n_{th} layer. In each step, the predictor $p_n^t(.)$ is optimized using the regressed anchor boxes $\mathcal{B}_n^{t-1}$ instead of the default anchor boxes. In other words, with the progressively refined anchor boxes, which means more positive samples could be available, the predictors in latter steps can be trained with a higher IoU threshold, which is helpful to produce more precise localization during inference [6]. Another advantage of this strategy is that multiple classifiers trained with different IoU thresholds in all steps will score each anchor box in a 'multi-expert' manner, and thus if properly fused the score will be more confident than a single classifier. Given this design, the limitations of current single-stage detectors could be alleviated, resulting in a potential of surpassing the two-stage detectors in both accuracy and efficiency.

Fig. 2 gives two example images to demonstrate the effectiveness of the proposed ALF module. As can be seen from Fig. 2 (a), there are only 7 and 16 default anchor boxes respectively assigned as positive samples under the IoU threshold of 0.5, this number increases progressively with more ALF steps, and the value of mean overlaps with the groundtruth is also going up. It indicates that the former predictor can hand over more anchor boxes with higher IoU to the latter one.

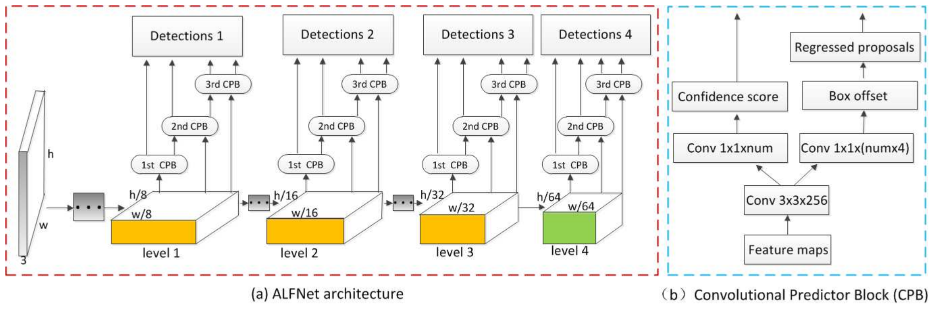


Fig. 3. (a) ALFNet architecture, which is constructed by four levels of feature maps for detecting objects with different sizes, where the first three blocks in yellow are from the backbone network, and the green one is an added convolutional layer to the end of the truncated backbone network. (b) Convolutional Predictor Block (CPB), which is attached to each level of feature maps to translate default anchor boxes to corresponding detection results.

3.3 Overall Framework

In this section we will present details of the proposed ALFNet pedestrian detection pipeline.

The details of our detection network architecture is pictorially illustrated in Fig. 3. Our method is based on a fully-convolutional network that produces a set of bounding boxes and confidence scores indicating whether there is a pedestrian instance or not. The base network layers are truncated from a standard network used for image classification (e.g. ResNet-50 [18] or MobileNet [20]). Taking ResNet-50 as an example, we firstly emanate branches from feature maps of the last layers of *stage 3, 4 and 5* (denoted as Φ_3, Φ_4 and Φ_5, the yellow blocks in Fig. 3 (a)) and attach an additional convolutional layer at the end to produce Φ_6, generating an auxiliary branch (the green block in Fig. 3 (a)). Detection is performed on $\{\Phi_3, \Phi_4, \Phi_5, \Phi_6\}$, with sizes downsampled by 8, 16, 32, 64 w.r.t. the input image, respectively. For proposal generation, anchor boxes with width of $\{(16, 24), (32, 48), (64, 80), (128, 160)\}$ pixels and a single aspect ratio of 0.41, are assigned to each level of feature maps, respectively. Then, we append the Convolutional Predictor Block (CPB) illustrated in Fig. 3 (b) with several stacked steps for bounding box classification and regression.

3.4 Training and Inference

Training Anchor boxes are assigned as positives S_+ if the IoUs with any ground truth are above a threshold u_h, and negatives S_- if the IoUs lower than a threshold u_l. Those anchors with IoU in $[u_l, u_h)$ are ignored during training. We assign different IoU threshold sets $\{u_l, u_h\}$ for progressive steps which will be discussed in our experiments.

At each step t, the convolutional predictor is optimized by a multi-task loss function combining two objectives:

$$L = l_{cls} + \lambda[y = 1]l_{loc}, \tag{7}$$

where the regression loss l_{loc} is the same smooth L1 loss adopted in Faster-RCNN [32], l_{cls} is cross-entropy loss for binary classification, and λ is a trade-off parameter. Inspired by [26], we also append the focal weight in classification loss l_{cls} to combat the positive-negative imbalance. The l_{cls} is formulated as:

$$l_{cls} = -\alpha \sum_{i \in S_+} (1 - p_i)^\gamma \log(p_i) - (1 - \alpha) \sum_{i \in S_-} p_i^\gamma \log(1 - p_i), \tag{8}$$

where p_i is the positive probability of sample i, α and γ are the focusing parameters, experimentally set as $\alpha = 0.25$ and $\gamma = 2$ suggested in [26]. In this way, the loss contribution of easy samples are down-weighted.

To increase the diversity of the training data, each image is augmented by the following options: after random color distortion and horizontal image flip with a probability of 0.5, we firstly crop a patch with the size of [0.3, 1] of the original image, then the patch is resized such that the shorter side has N pixels ($N = 640$ for CityPersons, and $N = 336$ for Caltech), while keeping the aspect ratio of the image.

Inference ALFNet simply involves feeding forward an image through the network. For each level, we get the regressed anchor boxes from the final predictor and hybrid confidence scores from all predictors. We firstly filter out boxes with scores lower than 0.01, then all remaining boxes are merged with the Non-Maximum Suppression (NMS) with a threshold of 0.5.

4 Experiments and Analysis

4.1 Experiment settings

Datasets. The performance of ALFNet is evaluated on the CityPersons [44] and Caltech [12] benchmarks. The CityPersons dataset is a newly published large-scale pedestrian detection dataset, which has 2975 images and approximately 20000 annotated pedestrian instances in the training subset. The proposed model is trained on this training subset and evaluated on the validation subset. For Caltech, our model is trained and test with the new annotations provided by [43]. We use the 10x set (42782 images) for training and the standard test subset (4024 images) for evaluation.

The evaluation metric follows the standard Caltech evaluation [12]: log-average Miss Rate over False Positive Per Image (FPPI) range of $[10^{-2}, 10^{0}]$ (denoted as MR^{-2}). Tests are only applied on the original image size without enlarging for speed consideration.

Training details. Our method is implemented in the Keras [7], with 2 GTX 1080Ti GPUs for training. A mini-batch contains 10 images per GPU. The

Adam solver is applied. For CityPersons, the backbone network is pretrained on ImageNet [9] and all added layers are randomly initialized with the xavier method. The network is totally trained for $240k$ iterations, with the initial learning rate of 0.0001 and decreased by a factor of 10 after $160k$ iterations. For Caltech, we also include experiments with the model initialized from CityPersons as done in [44,40] and totally trained for $140k$ iterations with the learning rate of 0.00001. The backbone network is ResNet-50 [18] unless otherwise stated.

4.2 Ablation Experiments

In this section, we conduct the ablation studies on the CityPersons validation dataset to demonstrate the effectiveness of the proposed method.

ALF improvement. For clarity, we trained a detector with two steps. Table 1 summarizes the performance, where C_iB_j represents the detection results obtained by the confidence scores on step i and bounding box locations on step j. As can be seen from Table 1, when evaluated with different IoU thresholds (e.g. 0.5, 0.75), the second convolutional predictor consistently performs better than the first one. With the same confidence scores C_1, the improvement from C_1B_2 to C_1B_1 indicates the second regressor is better than the first one. On the other hand, with the same bounding box locations B_2, the improvement from C_2B_2 to C_1B_2 indicates the second classifier is better than the first one.

We also combine the two confidence scores by summation or multiplication, which is denoted as $(C_1 + C_2)$ and $(C_1 * C_2)$. For the IoU threshold of 0.5, this kind of score fusion is considerably better than both C_1 and C_2. Yet interestingly, under a stricter IoU threshold of 0.75, both the two hybrid confidence scores underperform the second confidence score C_2, which reasonably indicates that the second classifier is more discriminative between groundtruth and many "close but not accurate" false positives. It is worth noting that when we increase the IoU threshold from 0.5 to a stricter 0.75, the largest improvement increases by a large margin (from 1.45 to 11.93), demonstrating the high-quality localization performance of the proposed ALFNet.

To further demonstrate the effectiveness of the proposed method, Fig. 4 depicts the distribution of anchor boxes over the IoU range of [0.5, 1]. The total number of matched anchor boxes increases by a large margin (from 16351 up to 100571). Meanwhile, the percentage of matched anchor boxes in higher IoU intervals is increasing stably. In other words, anchor boxes with different IoU values are relatively well-distributed with the progressive steps.

IoU threshold for training. As shown in Fig. 4, the number of matched anchor boxes increases drastically in latter steps, and the gap among different IoU thresholds is narrowing down. A similar finding is also observed in the Cascade R-CNN [6] with a single threshold, instead of dual thresholds here. This inspires us to study how the IoU threshold for training affects the final detection performance. Experimentally, the $\{u_l, u_h\}$ for the first step should not be higher than that for the second step, because more anchors with higher quality are assigned as positives after the first step (shown in Fig. 4). Results in Table 2 shows that training predictors of two steps with the increasing IoU thresholds is

Table 1. The ALF improvement evaluated under IoU threshold of 0.5 and 0.75. C_i represents the confidence scores from step i and B_j means the bounding box locations from step j. MR^{-2} on the reasonable subset is reported.

IoU	C_1B_1	C_1B_2	C_2B_2	$(C_1+C_2)B_2$	$(C_1*C_2)B_2$	Improvement
0.5	13.46	13.17	12.64	12.03	**12.01**	+1.45 (10.8%)
0.75	46.83	45.00	**34.90**	36.49	36.49	+11.93 (25.5%)

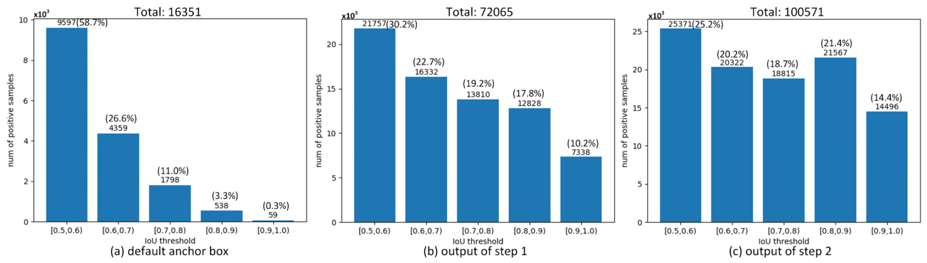


Fig. 4. It depicts the number of anchor boxes matched with the ground-truth boxes w.r.t. different IoU thresholds ranging from 0.5 to 1. (a), (b) and (c) represent the distribution of default anchor boxes, refined anchor boxes after the first and second step, respectively. The total number of boxes with IoU above 0.5 is presented in the heads of the three sub-figures. The numbers and percentages of each IoU threshold range are annotated on the head of the corresponding bar.

better than that with the same IoU thresholds, which indicates that optimizing the later predictor more strictly with higher-quality positive anchors is vitally important for better performance. We choose {0.3, 0.5} and {0.5, 0.7} for two steps in the following experiments, which achieves the lowest MR^{-2} in both of the two evaluated settings (IoU=0.5, 0.75).

Number of stacked steps. The proposed ALF module is helpful to achieve better detection performance, but we have not yet studied how many stacked steps are enough to obtain a speed-accuracy trade-off. We train our ALFNet up to three steps when the accuracy is saturated. Table 3 compares the three variants of our ALFNet with 1, 2 and 3 steps, denoted as ALFNet-1s, ALFNet-2s and ALFNet-3s. Experimentally, the ALFNet-3s is trained with IoU thresholds {0.3, 0.5}, {0.4, 0.65} and {0.5, 0.75}). By adding a second step, ALFNet-2s significantly surpasses ALFNet-1s by a large margin (12.01 VS. 16.01). It is worth noting that the results from the first step of ALFNet-2s and ALFNet-3s are substantially better than ALFNet-1s with the same computational burden, which indicates that multi-step training is also beneficial for optimizing the former step. Similar findings can also be seen in Cascade R-CNN [6], in which the three-stage cascade achieves the best trade-off.

From the results shown in Table 3, it appears that the addition of the 3rd step can not provide performance gain in terms of MR^{-2}. Yet when taking a

Table 2. Comparison of training the two-step ALFNet with different IoU threshold sets. $\{u_l, u_h\}$ represents the IoU threshold to assign positives and negatives defined in Section. 3.3. Bold and italic indicate the best and second best results.

Training IoU thresholds		MR^{-2}	
step 1	step 2	IoU=0.5	IoU=0.75
{0.3, 0.5}	{0.3, 0.5}	13.75	44.27
	{0.4, 0.6}	13.31	39.30
	{0.5, 0.7}	**12.01**	*36.49*
{0.4, 0.6}	{0.4, 0.6}	13.60	42.31
	{0.5, 0.7}	*12.80*	**36.43**
{0.5, 0.7}	{0.5, 0.7}	13.72	38.20

Table 3. Comparison of ALFNet with various steps evaluated in terms of MR^{-2}. Test time is evaluated on the original image size (1024x2048 on CityPersons).

Method	# Steps	Test step	Test time	MR^{-2}	
				IoU=0.5	IoU=0.75
ALFNet-1s	1	1	0.26s/img	16.01	48.95
ALFNet-2s	2	1	0.26s/img	13.17	45.00
	2	2	0.27s/img	**12.01**	**36.49**
ALFNet-3s	3	1	0.26s/img	14.53	46.70
	3	2	0.27s/img	12.67	37.75
	3	3	0.28s/img	12.88	39.31

deep look at the detection results of this three variants of ALFNet, the detection performance based on the metric of F-measure is further evaluated, as shown in Table 4. In this case, ALFNet-3s tested on the 3rd step performs the best under the IoU threshold of both 0.5 and 0.75. It substantially outperforms ALFNet-1s and achieves a 6.3% performance gain from ALFNet-2s under the IoU of 0.5, and 6.5% with IoU=0.75. It can also be observed that the number of false positives decreases progressively with increasing steps, which is pictorially illustrated in Fig. 5. Besides, as shown in Table 4, the average mean IoU of the detection results matched with the groundtruth is increasing, further demonstrating the improved detection quality. However, the improvement of step 3 over step 2 is saturating, compared to the large gap of step 2 over step 1. Therefore, considering the speed-accuracy trade-off, we choose ALFNet-2s in the following experiments.

Different backbone network. Large backbone network like ResNet-50 is *strong* in feature representation. To further demonstrate the improvement from the ALF module, a light-weight network like MobileNet [20] is chosen as the backbone and the results are shown in Table 5. Notably, the *weaker* MobileNet equipped with the proposed ALF module is able to beat the *strong* ResNet-50 without ALF (15.45 VS. 16.01).

Table 4. Comparison of ALFNet with various steps evaluated with F-measure. # TP and # FP denote the number of True Positives and False Positives.

Method	Test step	Ave. mIoU	IoU=0.5			IoU=0.75		
			# TP	# FP	F-mea.	# TP	# FP	F-mea.
ALFNet-1s	1	0.49	**2404**	13396	0.263	1786	14014	0.195
ALFNet-2s	1	0.55	2393	9638	0.330	**1816**	10215	0.250
	2	0.76	2198	1447	0.717	1747	1898	0.570
ALFNet-3s	1	0.57	2361	7760	0.375	1791	8330	0.284
	2	0.76	2180	1352	0.725	1734	1798	0.576
	3	0.80	2079	**768**	**0.780**	1694	**1153**	**0.635**

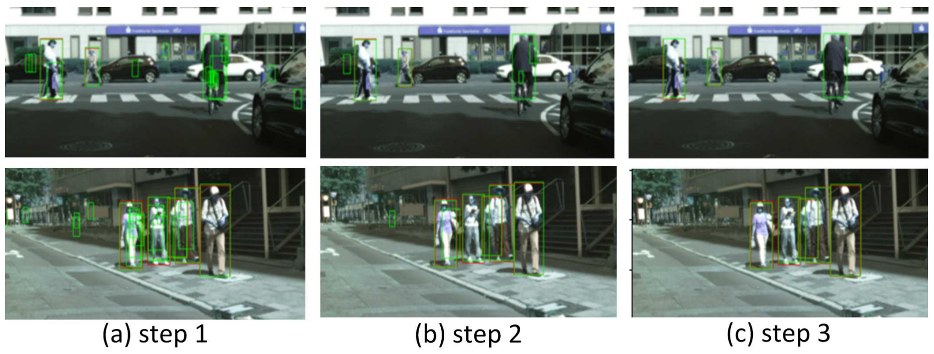


Fig. 5. Examples of detection results of ALFNet-3s. Red and green rectangles represent groundtruth and detection bounding boxes, respectively. It can be seen that the number of false positives decreases progressively with increasing steps, which indicates that more steps are beneficial for higher detection accuracy.

4.3 Comparison with State-of-the-Art

CityPersons. Table 6 shows the comparison to previous state-of-the art on CityPersons. Detection results test on the original image size are compared. Note that it is a common practice to upsample the image to achieve a better detection accuracy, but with the cost of more computational expense. We only test on the original image size as pedestrian detection is more critical on both accuracy and efficiency. Besides the reasonable subset, following [40], we also test our method on three subsets with different occlusion levels. On the Reasonable subset, without any additional supervision like semantic labels (as done in [44]) or auxiliary regression loss (as done in [40]), our method achieves the best performance, with an improvement of 1.2 MR^{-2} from the closest competitor RepLoss [40]. Note that RepLoss [40] is specifically designed for the occlusion problem, however, without bells and whistles, the proposed method with the same backbone network (ResNet-50) achieves comparable or

Table 5. Comparison of different backbone network with our ALF design.

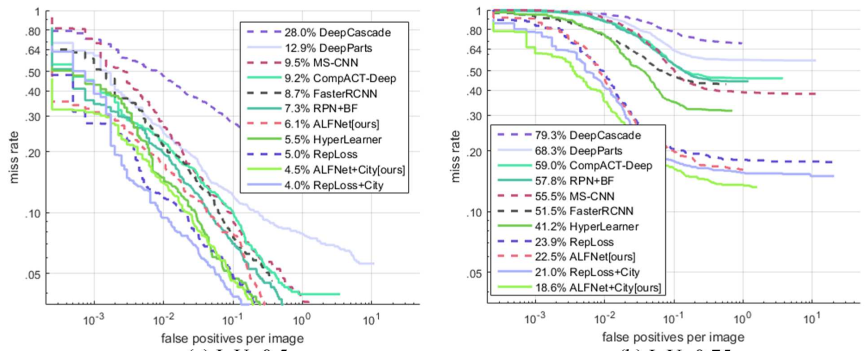
Backbone	Asymptotic Localization Fitting	# Parameters	MR^{-2}	
			IoU=0.5	IoU=0.75
ResNet-50		39.5M	16.01	48.94
	✓	48.4M	12.01	36.49
MobileNet		12.1M	18.88	56.26
	✓	17.4M	15.45	47.42

Table 6. Comparison with the state-of-the-art on the CityPersons[44]. Detection results test on the original image size (1024x2048 on CityPersons) is reported.

Method	+RepGT	+RepBox	+Seg.	Reasonable	Heavy	Partial	Bare
Faster-RCNN[44]				15.4	-	-	-
(VGG16)			✓	14.8	-	-	-
RepLoss[40] (ResNet-50)				14.6	60.6	18.6	7.9
	✓			13.7	57.5	17.3	**7.2**
		✓		13.7	59.1	17.2	7.8
	✓	✓		13.2	56.9	16.8	7.6
ALFNet[ours]				**12.0**	**51.9**	**11.4**	8.4

Table 7. Comparisons of running time on Caltech. The time of LDCF, CCF, CompACT-Deep and RPN+BF are reported in [42], and that of SA-FastRCNN and F-DNN are reported in [13]. MR^{-2} is based on the new annotations [43]. The original image size on Caltech is 480x640.

Method	Hardware	Scale	Test time	MR^{-2}	
				IoU=0.5	IoU=0.75
LDCF [29]	CPU	x1	0.6 s/img	23.6	72.2
CCF [41]	Titan Z GPU	x1	13 s/img	23.8	97.4
CompACT-Deep [5]	Tesla K40 GPU	x1	0.5 s/img	9.2	59.0
RPN+BF [42]	Tesla K40 GPU	x1.5	0.5 s/img	7.3	57.8
SA-FastRCNN [24]	Titan X GPU	x1.7	0.59 s/img	7.4	55.5
F-DNN [13]	Titan X GPU	x1	0.16 s/img	6.9	59.8
ALFNet [ours]	GTX 1080Ti GPU	x1	**0.05 s/img**	6.1	22.5
ALFNet+City [ours]	GTX 1080Ti GPU	x1	**0.05 s/img**	**4.5**	**18.6**



(a) IoU=0.5 (b) IoU=0.75

Fig. 6. Comparisons of state-of-the-arts on Caltech (reasonable subset).

even better performance in terms of different levels of occlusions, demonstrating the self-contained ability of our method to handle occlusion issues in crowded scenes. This is probably because in the latter ALF steps, more positive samples are recalled for training, including occluded samples. On the other hand, harder negatives are mined in the latter steps, resulting in a more discriminant predictor.

Caltech. We also test our method on Caltech and the comparison with state-of-the-arts on this benchmark is shown in Fig. 6. Our method achieves MR^{-2} of 4.5 under the IoU threshold of 0.5, which is comparable to the best competitor (4.0 of RepLoss [40]) . However, in the case of a stricter IoU threshold of 0.75, our method is the first one to achieve the MR^{-2} below 20.0%, outperforming all previous state-of-the-arts with an improvement of 2.4 MR^{-2} over RepLoss [40]. It indicates that our method has a substantially better localization accuracy.

Table 7 reports the running time on Caletch, our method significantly outperforms the competitors on both speed and accuracy. The speed of the proposed method is 20 FPS with the original 480x640 images. Thanks to the ALF module, our method avoids the time-consuming proposal-wise feature extraction (ROIpooling), instead, it refines the default anchors step by step, thus achieves a better speed-accuracy trade-off.

5 Conclusions

In this paper, we present a simple but effective single-stage pedestrian detector, achieving competitive accuracy while performing faster than the state-of-the-art methods. On top of a backbone network, an asymptotic localization fitting module is proposed to refine anchor boxes step by step into final detection results. This novel design is flexible and independent of any backbone network, without being limited by the single-stage detection framework. Therefore, it is also interesting to incorporate the proposed ALF module with other single-stage detectors like YOLO [30,31] and FPN [25,26], which will be studied in future.

References

1. Bell, S., Lawrence Zitnick, C., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2874–2883 (2016)
2. Benenson, R., Omran, M., Hosang, J., Schiele, B.: Ten years of pedestrian detection, what have we learned? In: European Conference on Computer Vision. pp. 613–627. Springer (2014)
3. Brazil, G., Yin, X., Liu, X.: Illuminating pedestrians via simultaneous detection & segmentation. arXiv preprint arXiv:1706.08564 (2017)
4. Cai, Z., Fan, Q., R.S.Feris, Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: European Conference on Computer Vision. pp. 354–370. Springer (2016)
5. Cai, Z., Saberian, M., Vasconcelos, N.: Learning complexity-aware cascades for deep pedestrian detection. international conference on computer vision pp. 3361–3369 (2015)
6. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. arXiv preprint arXiv:1712.00726 (2017)
7. Chollet, F.: Keras. published on github (https://github.com/fchollet/keras), (2015)
8. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. pp. 379–387 (2016)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. IEEE (2009)
10. Dollár, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(8), 1532–1545 (2014)
11. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features (2009)
12. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE transactions on pattern analysis and machine intelligence **34**(4), 743–761 (2012)
13. Du, X., El-Khamy, M., Lee, J., Davis, L.: Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection. In: Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on. pp. 953–961. IEEE (2017)
14. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
15. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware cnn model. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1134–1142 (2015)
16. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
17. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

19. Hosang, J., Omran, M., Benenson, R., Schiele, B.: Taking a deeper look at pedestrians. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4073–4082 (2015)
20. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
21. Kong, T., Sun, F., Yao, A., Liu, H., Lu, M., Chen, Y.: Ron: Reverse connection with objectness prior networks for object detection. arXiv preprint arXiv:1707.01691 (2017)
22. Kong, T., Yao, A., Chen, Y., Sun, F.: Hypernet: Towards accurate region proposal generation and joint object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 845–853 (2016)
23. Lee, H., Eum, S., Kwon, H.: Me r-cnn: multi-expert region-based cnn for object detection. arXiv preprint arXiv:1704.01069 (2017)
24. Li, J., Liang, X., Shen, S., Xu, T., Feng, J., Yan, S.: Scale-aware fast r-cnn for pedestrian detection. IEEE Transactions on Multimedia (2017)
25. Lin, Y.T., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. arXiv preprint arXiv:1612.03144 (2016)
26. Lin, Y.T., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. arXiv preprint arXiv:1708.02002 (2017)
27. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., S.Reed, Fu, C.Y., Berg, A.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
28. Mao, J., Xiao, T., Jiang, Y., Cao, Z.: What can help pedestrian detection? In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 1, p. 3 (2017)
29. Nam, W., Dollár, P., Han, J.H.: Local decorrelation for improved pedestrian detection. In: Advances in Neural Information Processing Systems. pp. 424–432 (2014)
30. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
31. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. arXiv preprint **1612** (2016)
32. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
33. Shen, Z., Liu, Z., Li, J., Jiang, Y.G., Chen, Y., Xue, X.: Dsod: Learning deeply supervised object detectors from scratch. In: The IEEE International Conference on Computer Vision (ICCV). vol. 3, p. 7 (2017)
34. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 761–769 (2016)
35. Shrivastava, A., Gupta, A.: Contextual priming and feedback for faster r-cnn. In: European Conference on Computer Vision. pp. 330–348. Springer (2016)
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
37. Tian, Y., Luo, P., Wang, X., Tang, X.: Deep learning strong parts for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision. pp. 1904–1912 (2015)

38. Tian, Y., Luo, P., Wang, X., Tang, X.: Pedestrian detection aided by deep learning semantic tasks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5079–5087 (2015)
39. Wang, X., Shrivastava, A., Gupta, A.: A-fast-rcnn: Hard positive generation via adversary for object detection. arXiv preprint arXiv:1704.03414 **2** (2017)
40. Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J., Shen, C.: Repulsion loss: Detecting pedestrians in a crowd. arXiv preprint arXiv:1711.07752 (2017)
41. Yang, B., Yan, J., Lei, Z., Li, S.Z.: Convolutional channel features. In: Computer Vision (ICCV), 2015 IEEE International Conference on. pp. 82–90. IEEE (2015)
42. Zhang, L., Lin, L., Liang, X., He, K.: Is faster r-cnn doing well for pedestrian detection? In: European Conference on Computer Vision. pp. 443–457. Springer (2016)
43. Zhang, S., Benenson, R., Omran, M., Hosang, J., Schiele, B.: How far are we from solving pedestrian detection? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1259–1267 (2016)
44. Zhang, S., Benenson, R., Schiele, B.: Citypersons: A diverse dataset for pedestrian detection. arXiv preprint arXiv:1702.05693 (2017)
45. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. arXiv preprint arXiv:1711.06897 (2017)