High-level Semantic Feature Detection: A New Perspective for Pedestrian Detection

Wei Liu[1,2]*, Shengcai Liao[3]†, Weiqiang Ren[4], Weidong Hu[1], Yinan Yu[4]
[1] ATR, College of Electronic Science, National University of Defense Technology, Changsha, China
[2] CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing ,China
[3] Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE
[4] Horizon Robotics Inc., Beijing ,China
{liuwei16, wdhu}@nudt.edu.cn, scliao@ieee.org, {weiqiang.ren, yinan.yu}@hobot.cc

Abstract

Object detection generally requires sliding-window classifiers in tradition or anchor-based predictions in modern deep learning approaches. However, either of these approaches requires tedious configurations in windows or anchors. In this paper, taking pedestrian detection as an example, we provide a new perspective where detecting objects is motivated as a high-level semantic feature detection task. Like edges, corners, blobs and other feature detectors, the proposed detector scans for feature points all over the image, for which the convolution is naturally suited. However, unlike these traditional low-level features, the proposed detector goes for a higher-level abstraction, that is, we are looking for central points where there are pedestrians, and modern deep models are already capable of such a high-level semantic abstraction. Besides, like blob detection, we also predict the scales of the pedestrian points, which is also a straightforward convolution. Therefore, in this paper, pedestrian detection is simplified as a straightforward center and scale prediction task through convolutions. This way, the proposed method enjoys an anchor-free setting. Though structurally simple, it presents competitive accuracy and good speed on challenging pedestrian detection benchmarks, and hence leading to a new attractive pedestrian detector. Code and models will be available at `https://github.com/liuwei16/CSP`.

1. Introduction

Feature detection is one of the most fundamental problems in computer vision. It is usually viewed as a low-level technique, with typical tasks including edge detection (e.g. Canny [4], Sobel [41]), corner (or interest point) detection (e.g. SUSAN [40], FAST [37]), and blob (or region of interest point) detection (e.g. LoG [25], DoG [31], MSER [33]). Feature detection is of vital importance to a variety of computer vision tasks ranging from image representation, image matching to 3D scene reconstruction, to name a few.

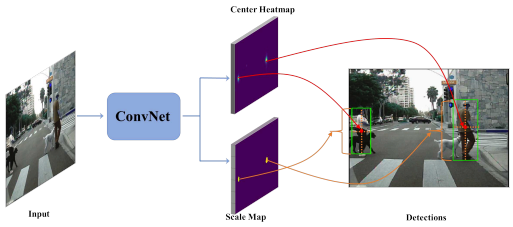

Figure 1. The overall pipeline of the proposed CSP detector. The final convolutions have two channels, one is a heatmap indicating the locations of the centers (red dots), and the other serves to predict the scales (yellow dotted lines) for each detected center.

Generally speaking, a feature is defined as an "interesting" part of an image, and so feature detection aims to compute abstractions of image information and make local decisions at every image point whether there is an image feature of a given type at that point or not [1]. Regarding abstraction of image information, with the rapid development for computer vision tasks, deep convolutional neural networks (CNN) are believed to be of very good capability to learn high-level image abstractions. Therefore, it has also been applied for feature detection, and demonstrates attractive successes even in low-level feature detections. For example, there is a recent trend of using CNN to perform edge detection [39, 47, 2, 29], which has substantially advanced this field. It shows that clean and continuous edges can be obtained by deep convolutions, which indicates that CNN has a stronger capability to learn higher-level abstraction of natural images than traditional methods. This capability may

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

not be limited to low-level feature detection; it may open up many other possibilities of high-level feature detection.

Therefore, in this paper, taking pedestrian detection as an example, we provide a new perspective where detecting objects is motivated as a high-level semantic feature detection task. Like edges, corners, blobs and other feature detectors, the proposed detector scans for feature points all over the image, for which the convolution is naturally suited. However, unlike these traditional low-level feature detectors, the proposed detector goes for a higher-level abstraction, that is, we are looking for central points where there are pedestrians. Besides, similar to the blob detection, we also predict the scales of the pedestrian points. However, instead of processing an image pyramid to determine the scale as in traditional blob detection, we predict object scale with also a straightforward convolution in one pass upon a fully convolution network (FCN) [30], considering its strong capability. As a result, pedestrian detection is simply formulated as a straightforward center and scale prediction task via convolution. The overall pipeline of the proposed method, denoted as **C**enter and **S**cale **P**rediction (**CSP**) based detector, is illustrated in Fig. 1.

As for general object detection, starting from the pioneer work of the Viola-Jones detector [45], it generally requires sliding-window classifiers in tradition or anchor-based predictions in CNN-based methods. These detectors are essentially local classifiers used to judge the pre-defined windows or anchors as being objects or not. However, either of these approaches requires tedious configurations in windows or anchors. Generally speaking, object detection is to tell where the object is, and how big it is. Traditional methods combines the "where" and "how" subproblems into a single one through the overall judgement of various scales of windows or anchors. In contrast, the proposed CSP detector separates the "where" and "how" subproblems into two different convolutions. This makes detection a more natural way, and enjoys a window-free or anchor-free setting, significantly reducing the difficulty in training.

There is another line of research which inspires us a lot. Previously, FCN has already been applied to and made a success in multi-person pose estimation [5, 34], where several keypoints are firstly detected merely through responses of full convolutions, and then they are further grouped into complete poses of individual persons. In view of this, recently two inspirational works, CornerNet [18] and TLL [42], successfully go free from windows and anchors, which perform object detection as convolutional keypoint detections and their associations. Though the keypoint association require additional computations, sometimes complex as in TLL, the keypoint prediction by FCN inspires us to go a step further, achieving center and scale prediction based pedestrian detection in full convolutions.

In summary, the main contributions of this work are as follows: (i) We show a new possibility that pedestrian detection can be simplified as a straightforward center and scale prediction task through convolutions, which bypasses the limitations of anchor-based detectors and gets rid of the complex post-processing of recent keypoint pairing based detectors. (ii) The proposed CSP detector achieves the new state-of-the-art performance on two challenging pedestrian detection benchmarks, CityPersons [51] and Caltech [9].

2. Related Works

2.1. Anchor-based object detection

One key component of anchor-based detectors is the anchor boxes of pre-defined scales and aspect ratios. In this way, detection is performed by classifying and regressing these anchor boxes. Faster R-CNN [36] is known as a two-stage detector, which generates objectness proposals and further classify and refine these proposals in a single framework. In contrast, singe-stage detectors, popularized by SSD [27], remove the proposal generation step and achieve comparable accuracy while are more efficient than two-stage detectors. In terms of pedestrian detection, Faster R-CNN has become the predominant framework. For example, RPN+BF [48] adapts the RPN and re-scores these proposals via boosted forests. MS-CNN [3] also applies the Faster R-CNN framework but generates proposals on multi-scale feature maps. Zhang *et al.* [51] contribute five strategies to adapt the plain Faster R-CNN for pedestrian detection. RepLoss [46] and OR-CNN [52] design two novel regression losses to tackle the occluded pedestrian detection in crowded scenes. Bi-Box [53] proposes an auxiliary subnetwork to predict the visible part of a pedestrian instance. Most recently, single-stage detectors also present competitive performance. For example, ALFNet [28] proposes the asymptotic localization fitting strategy to evolve the default anchor boxes step by step into precise detection results, and [21] focuses on the discriminative feature learning based on the original SSD architecture.

2.2. Anchor-free object detection

Anchor-free detectors bypass the requirement of anchor boxes and detect objects directly from an image. DeNet [44] proposes to generate proposals by predict the confidence of each location belonging to four corners of objects. Following the two-stage pipeline, DeNet also appends another subnetwork to re-score these proposals. Within the single-stage framework, YOLO [35] appends fully-connected layers to parse the final feature maps of a network into class confidence scores and box coordinates. Densebox [14] devises a unified FCN that directly regresses the classification scores and distances to the boundary of a ground truth box on all pixels, and demonstrates improved performance with landmark localization via multi-task learning. Most recently,

CornerNet [18] also applies a FCN but to predict objects' top-left and bottom-right corners and then group them via a associative embedding [34]. Enhanced by the novel corner pooling layer, CornerNet achieves superior performance on MS COCO object detection benchmark [22]. Similarly, TLL [42] proposes to detect an object by predicting the top and bottom vertexes. To group these paired keypoints into individual instances, it also predicts the link edge between them and employs a post-processing scheme based on Markov Random Field. Applying on pedestrian detection, TLL achieves significant improvement on Caltech [9], especially for small-scale pedestrians.

Our work also falls in the anchor-free object detection, but with significant differences to all above methods. We try to answer **to what extent a single FCN can be simplified for pedestrian detection**, and demonstrate that a single center point is feasible for object localization. Along with the scale prediction, CSP is able to generate bounding boxes without any requirements of extra post-processing schemes except the Non-Maximum Suppression (NMS).

2.3. Feature detection

Feature detection is a long-standing problem in computer vision with extensive literatures. Generally speaking, it mainly includes edge detection [4, 41], corner detection [37, 38], blob detection [33, 7] and so on. Traditional leading methods [4, 41] mainly focus on the utilization of local cues, such as brightness, colors, gradients and textures. With the development of CNN, a series of CNN-based method are proposed that significantly push forward the state of the arts in the task of feature detection. For example, there is a recent trend of using CNN to perform edge detection [39, 47, 2, 29], which have substantially advanced this field. However, different from these low-level feature points like edge, corners and blobs, the proposed method goes for a higher-level abstraction task, that is, we focus on detecting central points where there are pedestrians, for which modern deep models are already capable of.

3. Proposed Method

3.1. Preliminary

The CNN-based object detectors often rely on a backbone network (e.g. ResNet [12]). Taking an image I as input, the network may generate several feature maps with different resolutions, which can be defined as follows:

$$\phi_i = f_i(\phi_{i-1}) = f_i(f_{i-1}(...f_2(f_1(I)))), \tag{1}$$

where ϕ_i represents feature maps output by the ith layer. These feature maps decrease in size progressively and are generated by $f_i(.)$, which may be a combination of convolution or pooling, etc. Given a network with N layers, all the generated feature maps can be denoted as $\Phi = \{\phi_1, \phi_2, ..., \phi_N\}$, which is further utilized by detection heads.

Generally speaking, the CNN-based object detectors differ in how to utilize Φ. We denote these feature maps that are responsible for detection as Φ_{det}. In RPN [36], only the final feature map ϕ_N is used to perform detection, thus the final set of feature maps for detection is $\Phi_{det} = \{\phi_N\}$. While in SSD [27], the detection feature maps can be represented as $\Phi_{det} = \{\phi_L, \phi_{L+1}, ..., \phi_N\}$, where $1 < L < N$. Further, in order to enrich the semantic information of shallower layers for detecting small-scale objects, FPN [23] and DSSD [10] utilize the lateral connection to combine feature maps of different resolutions, resulting in $\Phi_{det} = \{\phi_L^{'}, \phi_{L+1}^{'}, ..., \phi_N^{'}\}$, where $\phi_i^{'}(i = L, L+1, ...N)$ is a combination of $\phi_i(i = L, L+1, ...N)$.

Besides Φ_{det}, in anchor-based detectors, another key component is called anchor boxes (denoted as $\mathcal{B}$). Given Φ_{det} and $\mathcal{B}$ in hand, detection can be formulated as:

$$\begin{aligned} Dets &= \mathcal{H}(\Phi_{det}, \mathcal{B}) \\ &= \{cls(\Phi_{det}, \mathcal{B}), regr(\Phi_{det}, \mathcal{B})\}, \end{aligned} \tag{2}$$

where $\mathcal{B}$ is pre-defined according to the corresponding set of feature maps Φ_{det}, and $\mathcal{H}(.)$ represents the detection head. Generally, $\mathcal{H}(.)$ contains two elements, namely $cls(.)$ which predicts the classification scores, and $regr(.)$ which predicts the scaling and offsets of the anchor boxes.

While in anchor-free detectors, detection is performed merely on the set of feature maps Φ_{det}, that is,

$$Dets = \mathcal{H}(\Phi_{det}) \tag{3}$$

3.2. Overall architecture

The overall architecture of the proposed CSP detector is illustrated in Fig. 2. The backbone network are truncated from a standard network pretrained on ImageNet [8] (e.g. ResNet-50 [12] and MobileNet [13]).

Feature Extraction. Taking ResNet-50 as an example, its *Conv* layers can be divided into five stages, in which the output feature maps are downsampled by 2, 4, 8, 16, 32 w.r.t. the input image. As a common practice [46, 42], the dilated convolutions are adopted in *stage 5* to keep its output as 1/16 of the input image size. We denote the output of *stage 2, 3, 4 and 5* as ϕ_2, ϕ_3, ϕ_4 and ϕ_5, in which the shallower feature maps can provide more precise localization information, while the coarser ones contain more semantic information with increasing the sizes of receptive fields. Therefore, we fuse these multi-scale feature maps from each stage into a single one in a simple way, that is, a deconvolution layer is adopted to make multi-scale feature maps with the same resolution before concatenation. Since the feature maps from each stage have different scales, we use L2-normalization to rescale their norms to 10,

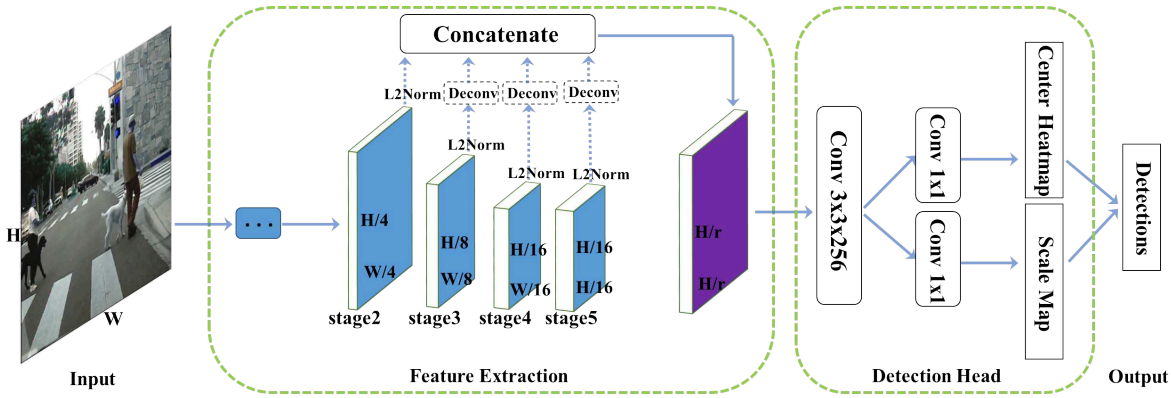

Figure 2. Overall architecture of CSP, which mainly comprises two components, i.e. the feature extraction module and the detection head. The feature extraction module concatenates feature maps of different resolutions into a single one. The detection head merely contains a 3x3 convolutional layer, followed by two prediction layers, one for the center location and the other for the corresponding scale.

which is similar to [21]. To investigate the optimal combination from these multi-scale feature maps, we conduct an ablative experiment in Sec. 4.2 and demonstrate that $\Phi_{det} = \{\phi_3, \phi_4, \phi_5\}$ is the best choice. Given an input image of size $H \times W$, the size of final concatenated feature maps is $H/r \times W/r$, where r is the downsampling factor. Similarly to [42], $r = 4$ gives the best performance as demonstrated in our experiments, because a larger r means coarser feature maps which struggle on accurate localization, while a smaller r brings more computational burdens. Note that more complicated feature fusion strategies like [23, 15, 17] can be explored to further improve the detection performance, but it is not in the scope of this work.

Detection Head. Upon the concatenated feature maps Φ_{det}, a detection head is appended to parse it into detection results. As stated in [26], the detection head plays a significant role in top performance, which has been extensively explored in the literature [10, 26, 20, 19]. In this work, we firstly attach a single 3x3 *Conv* layer on Φ_{det} to reduce its channel dimensions to 256, and then two sibling 1x1 *Conv* layers are appended to produce the center heatmap and scale map, respectively. Also, we do this for simplicity and any improvement of the detection head [10, 26, 20, 19] can be flexibly incorporate into this work to be a better detector.

A drawback from the downsampled feature maps is the problem of poor localization. Optionally, to slightly adjust the center location, an extra offset prediction branch can be appended in parallel with the above two branches.

3.3. Training

Ground Truth. The predicted heatmaps are with the same size as the concatenated feature maps (i.e. $H/r \times W/r$). Given the bounding box annotations, we can generate the center and scale ground truth automatically. An illustration example is depicted in Fig. 3 (b). For the center ground truth, the location where an object's center point falls is assigned as positive while all others are negatives.

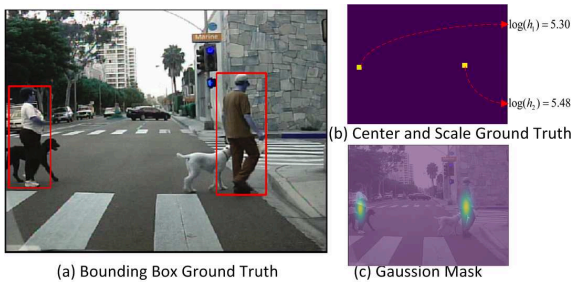

Figure 3. (a) is the bounding box annotations commonly adopted by anchor-based detectors. (b) is the center and scale ground truth generated automatically from (a). Locations of all objects' center points are assigned as positives, and negatives otherwise. Each pixel is assigned a scale value of the corresponding object if it is a positive point, or 0 otherwise. We only show the height information of the two positives for clarity. (c) is the overall Gaussian mask map M defined in Eq.4 to reduce the ambiguity of these negatives surrounding the positives.

Scale can be defined as the height and/or width of objects. Towards high-quality ground truth for pedestrian detection, line annotation is first proposed in [50, 51], where tight bounding boxes are automatically generated with a uniform aspect ratio of 0.41. In accordance to this annotation, we can merely predict the height of each object and generate the bounding box with the predetermined aspect ratio. For the scale ground truth, the kth positive location is assigned with the value of $log(h_k)$ corresponding to the kth object. To reduce the ambiguity, $log(h_k)$ is also assigned to the negatives within a radius 2 of the positives, while all other locations are assigned as zeros. Alternatively, we can also predict the width or height+width but with slightly poor performance for pedestrian detection as demonstrated in our

experiments (Sec. 4.2).

When the offset prediction branch is appended, the ground truth for the offsets of those centers can be defined as $(\frac{x_k}{r} - \lfloor \frac{x_k}{r} \rfloor, \frac{y_k}{r} - \lfloor \frac{y_k}{r} \rfloor)$.

Loss Function. For the center prediction branch, we formulate it as a classification task via the cross-entropy loss. Note that it is difficult to decide an 'exact' center point, thus the hard-designation of positives and negatives brings more difficulties for training. In order to reduce the ambiguity of these negatives surrounding the positives, we also apply a 2D Gaussian mask $G(.)$ centered at the location of each positive. An illustration example of the overall mask map M is depicted in Fig. 3 (c). Formally, it is formulated as:

$$\begin{aligned} M_{ij} &= \max_{k=1,2,\ldots,K} G(i,j;x_k,y_k,\sigma_{w_k},\sigma_{h_k}), \\ G(i,j;x,y,\sigma_w,\sigma_h) &= e^{-(\frac{(i-x)^2}{2\sigma_w^2}+\frac{(j-y)^2}{2\sigma_h^2})}, \end{aligned} \tag{4}$$

where K is the number of objects in an image, (x_k, y_k, w_k, h_k) is the center coordinates, width and height of the kth object, and the variances (σ_w^k, σ_h^k) of the Gaussian mask are proportional to the height and width of individual objects. If these masks have overlaps, we choose the maximum values for the overlapped locations. To combat the extreme positive-negative imbalance problem, the focal weights [24] on hard examples are also adopted. Thus, the classification loss can be formulated as:

$$L_{center} = -\frac{1}{K}\sum_{i=1}^{W/r}\sum_{j=1}^{H/r}\alpha_{ij}(1-\hat{p}_{ij})^{\gamma} log(\hat{p}_{ij}), \tag{5}$$

where

$$\begin{aligned} \hat{p}_{ij} &= \begin{cases} p_{ij} & \text{if } y_{ij}=1 \\ 1-p_{ij} & \text{otherwise,} \end{cases} \\ \alpha_{ij} &= \begin{cases} 1 & \text{if } y_{ij}=1 \\ (1-M_{ij})^{\beta} & \text{otherwise.} \end{cases} \end{aligned} \tag{6}$$

In the above, $p_{ij} \in [0,1]$ is the network's estimated probability indicating whether there is an object's center or not in the location (i,j), and $y_{ij} \in \{0,1\}$ specifies the ground truth label, where $y_{ij}=1$ represents the positive location. α_{ij} and γ are the focusing hyper-parameters, we experimentally set $\gamma=2$ as suggested in [24]. To reduce the ambiguity from those negatives surrounding the positives, the α_{ij} according to the Gaussian mask M is applied to reduce their contributions to the total loss, in which the hyper-parameter β controls the penalty. Experimentally, $\beta=4$ gives the best performance, which is similar to the one in [18]. For positives, α_{ij} is set as 1.

For scale prediction, we formulate it as a regression task via the smooth L1 loss [11]:

$$L_{scale} = \frac{1}{K}\sum_{k=1}^{K} SmoothL1(s_k, t_k), \tag{7}$$

where s_k and t_k represents the network's prediction and the ground truth of each positive, respectively.

If the offset prediction branch is appended, the similar smooth L1 loss in Eq. 7 is adopted (denoted as L_{offset}).

To sum up, the full optimization objective is:

$$L = \lambda_c L_{center} + \lambda_s L_{scale} + \lambda_o L_{offset}, \tag{8}$$

where λ_c, λ_s and λ_o are the weights for center classification, scale regression and offset regression losses, which are experimentally set as 0.01, 1 and 0.1, respectively.

Data Augmentation. To increase the diversity of the training data, standard data augmentation techniques are adopted. Firstly, random color distortion and horizontal flip are applied, followed by randomly scaled in the range of [0.4, 1.5]. Secondly, a patch is cropped or expanded by zero-padding such that the shorter side has a fixed number of pixels (640 for CityPersons [51], and 336 for Caltech [9]). Note that the aspect ratio of the image is kept during this process.

3.4. Inference

During testing, CSP simply involves a single forward of FCN with several predictions. Specifically, locations with confidence score above 0.01 in the center heatmap are kept, along with their corresponding scale in the scale map. Then bounding boxes are generated automatically and remapped to the original image size, followed by NMS with a threshold of 0.5. If the offset prediction branch is appended, the centers are adjusted accordingly before remapping.

4. Experiments

4.1. Experiment settings

Datasets. To demonstrate the effectiveness of the proposed method, we evaluate on two of the largest pedestrian detection benchmarks, i.e. Caltech [9] and CityPersons [51]. Caltech comprises approximately 2.5 hours of autodriving video with extensively labelled bounding boxes. Following [51, 32, 46, 28, 52], we use the training data augmented by 10 folds (42782 frames) and test on the 4024 frames in the standard test set, all experiments are conducted on the new annotations provided by [49]. CityPersons is a more challenging large-scale pedestrian detection dataset with various occlusion levels. We train the models on the official training set with 2975 images and test on the validation set with 500 images.

One reason we choose these two datasets lies in that they provide bounding boxes via central body line annotation and normalized aspect ratio, this annotation procedure is helpful to ensure the boxes align well with the centers of pedestrians. Evaluation follows the standard Caltech evaluation metric [9], that is log-average Miss Rate over False Positive Per Image (FPPI) ranging in $[10^{-2}, 10^{0}]$ (denoted

Point Prediction	$MR^{-2}(\%)$	
	IoU=0.5	IoU=0.75
Center point	**4.62**	**36.47**
Top vertex	7.75	44.70
Bottom vertex	6.52	40.25

Table 1. Comparisons of different high-level feature points. Bold number indicates the best result.

Scale Prediction	$MR^{-2}(\%)$	
	IoU=0.5	IoU=0.75
Height	**4.62**	**36.47**
Width	5.31	53.06
Height+Width	4.73	41.09

Table 2. Comparisons of different definitions for scale prediction. Bold number indicates the best result.

Disturbance (pixels)	$MR^{-2}(\%)$	$\Delta MR^{-2}(\%)$
0	4.62	-
[0, 4]	5.68	$\downarrow 1.06$
[0, 8]	8.59	$\downarrow 3.97$

Table 6. Performance drop with disturbances of the centers.

as MR^{-2}). Tests are only applied on the original image size without enlarging for speed consideration.

Training details. We implement the proposed method in Keras [6]. The backbone is ResNet-50 [12] pretrained on ImageNet [8] unless otherwise stated. Adam [16] is applied to optimize the network. We also apply the strategy of moving average weights proposed in [43] to achieve more stable training. For Caltech [9], a mini-batch contains 16 images with one GPU (GTX 1080Ti), the learning rate is set as 10^{-4} and training is stopped after 15K iterations. Following[51, 46, 28, 52], we also include experiments with the model initialized from CityPersons [51], which is trained with the learning rate of 2×10^{-5}. For CityPersons [51], we optimize the network on 4 GPUs with 2 images per GPU for a mini-batch, the learning rate is set as 2×10^{-4} and training is stopped after 37.5K iterations.

4.2. Ablation Study

In this section, an ablative analysis of the proposed method is conducted on the Caltech dataset, evaluations are based on the new annotations provided by [49].

Why is the Center Point? As a kind of high-level feature point, the center point is capable of locating an individual object. A question comes in that how about other high-level feature points. To answer this, we choose two other high-level feature points as adopted in [42], i.e. the top and bottom vertexes. Comparisons are reported in Table. 1. It is shown that both the two vertexes can succeed in detection but underperform the center point by approximately 2%-3% under IoU=0.5, and the performance gap is even larger under the stricter IoU=0.75. This is probably because the center point is advantageous to perceive the full body information and thus is easier for training.

How important is the Scale Prediction? Scale prediction is another indespensible component for bounding box generation. In practice, we merely predict the height for each detected center in accordance to the line annotation in [50, 51]. To demonstrate the generality of CSP, we have also tried to predict Width or Height+Width for comparison. For Height+Width, the only difference in network architecture lies in that the scale prediction branch has two channels responsible for the height and width respectively. It can be observed in Table 2 that Width and Height+Width prediction can also achieve comparable but suboptimal results to Height prediction. This result may be attributed to the line annotation adopted in [50, 51] which provides accurate height information with less noise during training. Besides, the ground truth for width is automatically generated by the annotated height information, thus is not able to provide additional information for training. With the comparable performance from Height +Width prediction, it makes CSP potentially feasible for other object detection tasks requiring both height and width.

How important is the Feature Resolution? In the proposed method, the final set of feature maps (denoted as Φ^r_{det}) is downsampled by r w.r.t the input image. To explore the influence from r, we train the models with $r = 2, 4, 8, 16$ respectively. For $r = 2$, Φ^2_{det} are upsampled from Φ^4_{det} by deconvolution. To remedy the issue of poor localization from downsampling, the offset prediction branch is alternatively appended for $r = 4, 8, 16$ to adjust the center location. Evaluations under IoU=0.75 are included to verify the effectiveness of additional offset prediction when stricter localization quality is required. As can be seen from Table. 3, without offset prediction, Φ^4_{det} presents the best result under IoU=0.5, but performs poorly under IoU=0.75 when compared with Φ^2_{det}, which indicates that finer feature maps are beneficial for precise localization. Though Φ^2_{det} performs the best under IoU=0.75, it does not bring performance gain under IoU=0.5 though with more computational burdens. Not surprisingly, a larger r witnesses a significant performance drop, which is mainly due to that coarser feature maps lead to poor localization. In this case, the offset prediction plays a significant role. Notably, additional offset prediction can substantially improve the detector upon Φ^{16}_{det} by 12.86% and 41.30% under the IoU threshold of 0.5 and 0.75, respectively. It can also achieve an improvement of 7.67% under IoU=0.75 for the detector upon Φ^4_{det}, even though the performance gain is saturating under IoU=0.5. It is worth noting that the extra computation cost from the offset prediction is negligible,

Feature for Detection	+Offset	Test Time (ms/img)	$MR^{-2}(\%)$		$\Delta MR^{-2}(\%)$	
			IoU=0.5	IoU=0.75	IoU=0.5	IoU=0.75
Φ^2_{det}		69.8	5.32	30.08	-	-
Φ^4_{det}		58.2	4.62	36.47	+0.08	+7.67
	✓	59.6	**4.54**	**28.80**		
Φ^8_{det}		49.2	7.00	54.25	+0.92	+21.32
	✓	50.4	6.08	32.93		
Φ^{16}_{det}		42.0	20.27	75.17	**+12.86**	**+41.30**
	✓	42.7	7.41	33.87		

Table 3. Comparisons of different downsampling factors of the feature maps, which are denoted as Φ^r_{det} downsampled by r w.r.t the input image. Test time is evaluated on the image with size of 480x640 pixels. ΔMR^{-2} means the improvement from the utilization of the offset prediction. Bold numbers indicate the best result.

Feature Maps				ResNet-50[12]			MobileNetV1[13]		
ϕ_2	ϕ_3	ϕ_4	ϕ_5	# Parameters	Test Time	$MR^{-2}(\%)$	# Parameters	Test Time	$MR^{-2}(\%)$
✓	✓			4.7MB	36.2ms/img	9.96	2.1MB	27.3ms/img	34.96
	✓	✓		16.1MB	44.5ms/img	5.68	6.0MB	32.3ms/img	**8.33**
		✓	✓	37.4MB	54.4ms/img	5.84	10.7MB	34.5ms/img	10.03
✓	✓	✓		16.7MB	46.0ms/img	6.34	6.3MB	33.3ms/img	8.43
	✓	✓	✓	40.0MB	58.2ms/img	**4.62**	12.3MB	38.2ms/img	9.59
✓	✓	✓	✓	40.6MB	61.1ms/img	4.99	12.6MB	40.5ms/img	9.05

Table 4. Comparisons of different combinations of multi-scale feature representations defined in Sec. 3.2. ϕ_2, ϕ_3, ϕ_4 and ϕ_5 represent the output of *stage 2, 3, 4 and 5* of a backbone network, respectively. Bold numbers indicate the best results.

with approximately 1ms per image of 480x640 pixels.

How important is the Feature Combination? It has been revealed in [42] that multi-scale representation is vital for pedestrian detection of various scales. In this part, we conduct an ablative experiment to study which combination of the multi-scale feature maps from the backbone is the optimal one. As the much lower layer has limited discriminant information, in practice we choose the output of stage 2 (ϕ_2) as a start point and the downsampling factor r is fixed as 4. In spite of the ResNet-50[12] with *stronger* feature representation, we also choose a light-weight network like MobileNetV1[13] as the backbone. The results in Table 4 shows that the much shallower feature maps like ϕ_2 result in poorer accuracy, while deeper feature maps like ϕ_4 and ϕ_5 are of great importance for superior performance, and the middle-level feature maps ϕ_3 are indispensable to achieve the best results. For ResNet-50, the best performance comes from the combination of $\{\phi_3, \phi_4, \phi_5\}$, while $\{\phi_3, \phi_4\}$ is the optimal one for MobileNetV1.

4.3. Comparison with the State of the Arts

Caltech. The proposed method are extensively compared with the state of the arts on three settings: Reasonable, All and Heavy Occlusion. As shown in Fig. 4, CSP achieves MR^{-2} of 4.5% on the Reasonable setting, which outperforms the best competitor (5.0 of RepLoss [46]) by 0.4%. When the model is initialized from CityPersons[51], CSP also achieves a new state of the art of 3.8%, compared to 4.0% of RepLoss [46], 4.1% of OR-CNN [52], and 4.5% of ALFNet [28]. It presents the superiority on detecting pedestrians of various scales and occlusion levels as demonstrated in Fig . 4 (b). Moreover, Fig. 4 (c) shows that CSP also performs very well for heavily occluded pedestrians, outperforming RepLoss [46] and OR-CNN [52] which are explicitly designed for occlusion cases.

CityPersons. Table 5 shows the comparisons with previous state of the arts on CityPersons. Besides the reasonable subset, following [46], we also evaluate on three subsets with different occlusion levels, and following [51], results on three subsets with various scale ranges are also included. It can be observed that CSP beats the competitors and performs fairly well on occlusion cases even without any specific occlusion-handling strategies [46, 52]. On the Reasonable subset, CSP with offset prediction achieves the best performance, with a gain of 1.0% MR^{-2} upon the closest competitor (ALFNet [28]), while the speed is comparable on the same running environment with 0.33 second per image of 1024x2048 pixels.

4.4. Discussions

Note that CSP only requires object centers and scales for training, though generating them from bounding box or central line annotations is more feasible since centers are not always easy to annotate. Besides, the model may be puzzled

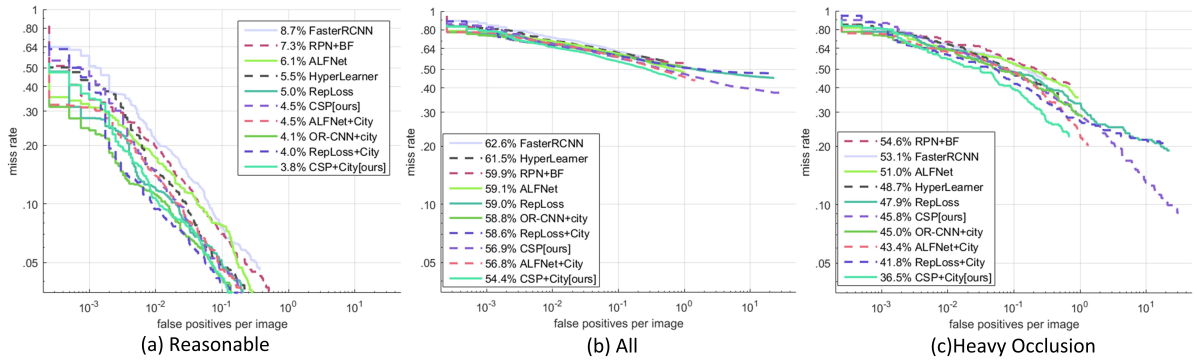

Figure 4. Comparisons with the state of the arts on Caltech using new annotations.

Method	Backbone	Reasonable	Heavy	Partial	Bare	Small	Medium	Large	Test Time
FRCNN[51]	VGG-16	15.4	-	-	-	25.6	7.2	7.9	-
FRCNN+Seg[51]	VGG-16	14.8	-	-	-	22.6	6.7	8.0	-
OR-CNN[52]	VGG-16	12.8	55.7	15.3	6.7	-	-	-	-
RepLoss[46]	ResNet-50	13.2	56.9	16.8	7.6	-	-	-	-
TLL[42]	ResNet-50	15.5	53.6	17.2	10.0	-	-	-	-
TLL+MRF[42]	ResNet-50	14.4	52.0	15.9	9.2	-	-	-	-
ALFNet[28]	ResNet-50	12.0	51.9	11.4	8.4	19.0	5.7	6.6	0.27s/img
CSP(w/o offset)	ResNet-50	11.4	49.9	10.8	8.1	18.2	3.9	6.0	0.33s/img
CSP(with offset)	ResNet-50	11.0	49.3	10.4	7.3	16.0	3.7	6.5	0.33s/img

Table 5. Comparison with the state of the arts on CityPersons[51]. Results test on the original image size (1024x2048 pixels) are reported. Red and green indicate the best and second best performance.

on ambiguous centers during training. To demonstrate this, we randomly disturbed object centers in the range of [0,4] and [0,8] pixels during training. From the results shown in Table 6, it can be seen that performance drops with increasing annotation noise. For Caltech, we also apply the original annotations but with inferior performance to TLL [42], which is also anchor-free. A possible reason is that TLL includes a series of post-processing strategies in keypoint pairing. For evaluation with tight annotations based on central lines, as results of TLL on Caltech are not reported in [42], comparison to TLL is given in Table 5 on the CityPersons, which shows the superiority of CSP. Therefore, the proposed method may be limited for annotations with ambiguous centers, e.g. the traditional pedestrian bounding box annotations affected by limbs. In view of this, it may also be not straightforward to apply CSP to generic object detection without further improvement or new annotations.

When compared with anchor-based methods, the advantage of CSP lies in two aspects. Firstly, CSP does not require tedious configurations on anchors specifically for each dataset. Secondly, anchor-based methods detect objects by overall classifications of each anchor where background information and occlusions are also included and will confuse the detector's training. However, CSP overcomes this drawback by scanning for pedestrian centers instead of boxes in an image, thus is more robust to occluded objects.

5. Conclusion

Inspired from the traditional feature detection task, we provide a new perspective where pedestrian detection is motivated as a high-level semantic feature detection task through straightforward convolutions for center and scale predictions. This way, the proposed method enjoys anchor-free settings and is also free from complex post-processing strategies as in recent keypoint-pairing based detectors. As a result, the proposed CSP detector achieves the new state-of-the-art performance on two challenging pedestrian detection benchmarks, namely CityPersons and Caltech. Due to the general structure of the CSP detector, it is interesting to further explore its capability in other tasks like face detection, vehicle detection, and general object detection.

Acknowledgment

This work was partly supported by the National Key Research and Development Plan (Grant No.2016YFC0801003), the NSFC Project #61672521, the NLPR Independent Research Project #Z-2018008, and the IIAI financial support.

References

[1] https://en.wikipedia.org/wiki/Feature_detection_(computer_vision).

[2] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4380–4389. IEEE, 2015.

[3] Z. Cai, Q. Fan, R.S.Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, pages 354–370. Springer, 2016.

[4] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.

[6] Franois Chollet. Keras. published on github (https://github.com/fchollet/keras),. 2015.

[7] Hongli Deng, Wei Zhang, Eric Mortensen, Thomas Dietterich, and Linda Shapiro. Principal curvature-based region detector for object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[8] J. Deng, W. Dong, R. Socher, L.-J Li, K. Li, and F.-F Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[9] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2012.

[10] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[11] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[14] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

[15] Seung-Wook Kim, Hyong-Keun Kook, Jee-Young Sun, Mun-Cheon Kang, and Sung-Jea Ko. Parallel feature pyramid network for object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Tao Kong, Fuchun Sun, Chuanqi Tan, Huaping Liu, and Wenbing Huang. Deep feature pyramid reconfiguration for object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[18] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[19] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. Dsfd: Dual shot face detector. *arXiv preprint arXiv:1810.10220*, 2018.

[20] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017.

[21] Chunze Lin, Jiwen Lu, Gang Wang, and Jie Zhou. Graininess-aware deep feature learning for pedestrian detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Y.-T Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. *arXiv preprint arXiv:1612.03144*, 2016.

[24] Y.-T Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

[25] Tony Lindeberg. Scale selection properties of generalized scale-space interest point detectors. *Journal of Mathematical Imaging and vision*, 46(2):177–210, 2013.

[26] Songtao Liu, Di Huang, and andYunhong Wang. Receptive field block net for accurate and fast object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.Reed, C.-Y Fu, and A.C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[28] Wei Liu, Shengcai Liao, Weidong Hu, Xuezhi Liang, and Xiao Chen. Learning efficient single-stage pedestrian detectors by asymptotic localization fitting. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[29] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5872–5881. IEEE, 2017.

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[31] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[32] Jiayuan Mao, Tete Xiao, Yuning Jiang, and Zhimin Cao. What can help pedestrian detection? In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017.

[33] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[34] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017.

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[36] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[37] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[38] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):105–119, 2010.

[39] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2015.

[40] Stephen M Smith and J Michael Brady. Susan: a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[41] Irwin Sobel. Camera models and machine perception. Technical report, Computer Science Department, Technion, 1972.

[42] Tao Song, Leiyu Sun, Di Xie, Haiming Sun, and Shiliang Pu. Small-scale pedestrian detection based on topological line localization and temporal feature aggregation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[43] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[44] Lachlan Tychsen-Smith and Lars Petersson. Denet: Scalable real-time object detection with directed sparse sampling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 428–436, 2017.

[45] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[46] Xinlong Wang, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. Repulsion loss: Detecting pedestrians in a crowd. *arXiv preprint arXiv:1711.07752*, 2017.

[47] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, 125(1-3):3–18, 2017.

[48] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, pages 443–457. Springer, 2016.

[49] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. How far are we from solving pedestrian detection? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1259–1267, 2016.

[50] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Towards reaching human performance in pedestrian detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):973–986, 2018.

[51] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. *arXiv preprint arXiv:1702.05693*, 2017.

[52] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Occlusion-aware r-cnn: Detecting pedestrians in a crowd. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[53] Chunluan Zhou and Junsong Yuan. Bi-box regression for pedestrian detection and occlusion estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.