IJICTR
International Journal of Information & Communication Technology Research

Volume 10- Number 4 – Autumn 2018 (42 -52)

Deep Learning Based on Parallel CNNs for Pedestrian Detection

Mahmoud Saeidi*
Faculty of Computer Engineering
K. N. Toosi University of Technology
Tehran, Iran
msaeidi40@itrc.ac.ir

Ali Ahmadi
Faculty of Computer Engineering
K. N. Toosi University of Technology
Tehran, Iran
ahmadi@eetd.kntu.ac.ir

Received: 9 May 2018 - Accepted: 2 September 2018

Abstract— **Recently, deep learning methods, mostly algorithms based on Deep Convolutional Neural Networks (DCNNs) have yielded great results on pedestrian detection. Algorithms based on DCNNs spontaneously learn features in a supervised manner and are able to learn qualified high level feature representations to detect pedestrian. In this paper, we first review a number of popular DCNN-based training approaches along with their recent extensions. We then briefly describe recent algorithms based on these approaches. Also, we accentuate recent contributions and main challenges of DCNNs in detecting pedestrian. We analyze deep pedestrian detection algorithms from training approach, categorization, and DCNN model points of view, and ultimately propose a new deep architecture and training approach for deep pedestrian detection. The experimental results show that the proposed DCNN and training approach, achieve more accurate rate detection than the previously reported architectures and training approaches.**

Keywords- Parallel DCNN; Pedestrian Detection; Region-based Convolutional Neural Network (RCNN); Single Shot Detector (SSD); Training Approach.

I. INTRODUCTION

Pedestrian detection is one of the most significant elements of wide ranges of applications such as automotive safety, robotics, self-driving car, pedestrian protection systems, and intelligent video surveillance. The main challenges of pedestrian detection are as follows:

- Geometric shape of pedestrian being similar to objects such as trees, statues and pylons.
- Pedestrian appearing in various clothing colors
- Variation of background scene
- Changing pedestrian poses
- Occlusion
- Huge and complicated Computing

To capture the most efficient information of pedestrian, we can utilize SIFT [1], HOG [2], and Haar-like features [3]. The deformable part-based models [4] detect human parts to address changeable geometric shapes. Occlusion can be managed by determining occluded regions of pedestrian [5, 6, 7, 8]. Training DCNN-based pedestrian detection algorithms requires massive amount of training data. Therefore, it calls for an impressively high performance computing model for both forward and backward passes. To alleviate the computational load, a number of methods have been proposed: Sharing features across multi-scale models [9], Markov Chain Monte Carlo [10], Parallel Ada-Boost Algorithm [11], suitable features and classifiers [12], GPU and CPU cooperation [13], and Multi-CPU and multi-GPU [14].

Recent researches focus on methods based on Deep Convolutional Neural Networks (DCNNs) to handle the challenges of pedestrian detection. DCNNs can model high-level abstractions in data by employing hierarchical architectures. In fact, DCNNs have

* Corresponding Author

recently illustrated the most promising performance on pedestrian detection.

Although the features extracted by pre-trained deep models may not be better than the traditional hand-crafted features, by proper feature refining schemes, DCNN feature representations consistently outperform hand-crafted features in pedestrian detection. For example, algorithms based on hand-crafted features and shallow learning such as SquaresChnFtrs [15], InformedHaar [16], and Katamari [17] perform better than algorithms based on deep models such as MultiSDP [18] and SDN [19] in pedestrian detection. On the other hand, recent algorithms based on deep models such as CompAct-Deep [20], DeepParts [5], and TA-CNN [21] are more efficient than SquaresChnFtrs, InformedHaar, and Katamari.

There are three main approaches for object detection using DCNN: Sliding-Window-based Convolutional Neural Networks (SWCNN) approach [22, 23], Region-based Convolutional Neural Network (RCNN) family approaches [24], and Single Shot Detector (SSD) approach [25]. All three can also be employed in pedestrian detection algorithms.

SWCNN approach makes use of multi-scale and sliding window to extract features using DCNN. The huge and complicated computational cost of SWCNN approach makes it less desirable as it requires processing of many image patches to generate appropriate bounding-box for detecting pedestrian.

RCNN-based family approaches are divided into three important approaches: RCNN, Fast RCNN, and Faster RCNN. The first approach, RCNN utilizes the results of a proper Object Proposal Algorithm (OPA) to extract feature maps in DCNN for classification and bounding-box regression. Although OPAs are suitable at finding object positions, they are not able to perform accurate localization of the entire object by a tight bounding-box. Another prominent approach is Fast RCNN [26] in which the entire image and the extracted candidate objects are considered as inputs to the DCNN model. The last and high performance approach of RCNN family, Faster RCNN [27] utilizes a Region Proposal Network (RPN) to share full-image convolutional features with the detection network. RPN is a fully convolutional network that predicts object-bounds and object-ness scores all together at each position. In short, RCNN approach is improved in [26, 27] from accuracy and speed points of view by make use of sharing computation.

SSD approach partitions the output space of bounding-boxes into a set of default boxes over a variety of aspect ratios and scales per feature map location. it thoroughly omits proposal generation and subsequent pixel or feature resampling stages and encloses all computation in a single network.

Recently, a number of methods based on DCNNs achieve great performance in detecting pedestrian [19, 20, 5, 21, 28, 9, 29]. The state-of-the-art DCNN-based pedestrian detection algorithms are distinct from each other's, and also each of them consists of a definitive innovation in extracting feature maps to detect pedestrian. SWCNN, RCNN, and SSD approaches have been utilized to train and fine-tune DCNN parameters in deep pedestrian detection.

To evaluate pedestrian detection algorithms, different pedestrian datasets have been developed. The most extensively used datasets includes Caltech-USA [30], INRIA [2], KITTI [31], and ETH [32]. Caltech-USA and KITTI are the most challenging benchmarks with comparatively extensive data. INRIA is the oldest dataset which covers pedestrian in various environment (street, beach, mountains, etc.). ETH is a mid-sized benchmark and provides stereo information.

In our contribution, we first propose a novel architecture based on parallel DCNNs. Then, we propose a new training approach based on Faster RCNN to detect pedestrian. The proposed approach is superior to the previous ones from accuracy point of view.

II. Training Approaches for Pedestrian Detection based on DCNN

Training approaches used in DCNN-based object detection algorithms can be employed for pedestrian detection. There are three prominent DCNN-based approaches in object detection: SWCNN, RCNN, and SSD which can also be utilized for pedestrian detection algorithms. In this section, we overview and analyze these approaches from training, accuracy and speed points of view.

A. Pedestrian Detection based on SWCNN Approach

SWCNN approach employs multi-scale sliding window to extract and process features. It then classifies and localizes objects using DCNNs. In other words, this approach can localize pedestrian without object proposal algorithms. The main challenge of SWCNN approach is to detect potentially multiple pedestrians with different sizes within the same image using a finite amount of computing resources.

There are two main strategies in SWCNN approach for pedestrian detection: the first one is to keep DCNN unaltered and simultaneously resize input to the input size of DCNN. The second strategy is to keep the input image unaltered while utilizing multiple sizes in DCNN. The first strategy confines both the aspect ratio and the scale of the input. Although the second strategy has no this restriction, it is slower than the first strategy in training and fine-tuning DCNN.

In short, the heavy and complicated computational cost of SWCNN approach makes it less desirable as it requires processing several image patches in order to generate appropriate bounding-box in pedestrian detection.

B. RCNN Approaches Family for Pedestrian Detection

Pedestrian detection based on RCNN approach is slow. To address this drawback, the improved RCNN approaches, Fast RCNN [26] and Faster RCNN [27] share computation to accelerate processing, and increase accuracy in detecting object.

RCNN Approach for Pedestrian Detection- RCNN employs the results of OPAs such as object-ness of image windows [33], selective search [34], category-

independent object proposals [35], BING [36], and edge boxes [37] to extract features maps to train and fine-tune classifiers and bounding-box regression. In RCNN approach, The OPAs utilized in pedestrian detection algorithms are distinct from the OPAs utilized in general object detection. Although the selective search method is one of the most popular OPAs for generic object detection, it is not effective in pedestrian detection algorithms. In fact, the recall of ground-truth annotations degrades severely when the IoU threshold increases. It means that selective search immediately denies a large number of pedestrian candidates which results in a high miss rate regardless of pedestrian detection algorithms performance. Moreover, OPAs such as SquaresChnFtrs [15] and Katamari-v1 [17] yield far better candidate pedestrians than the generic OPAs such as selective search [34] and BING [36]. As shown in Fig. 1, RCNN approach detects pedestrian in five stages. At the first stage, pedestrian candidates are specified using an appropriate OPA such as SquaresChnFtrs. Secondly, a fixed-length feature vector is extracted from each warped pedestrian proposal using DCNN. Next, a linear Support-Vector Machine (SVM) is trained to optimize pedestrian detection. At the fourth stage, bounding-box regression is performed to locate pedestrian accurately. Finally, at the last stage, scores of bounding-boxes are ranked and Non-Maximum Suppression (NMS) is employed to select the final bounding-boxes as detected pedestrians.

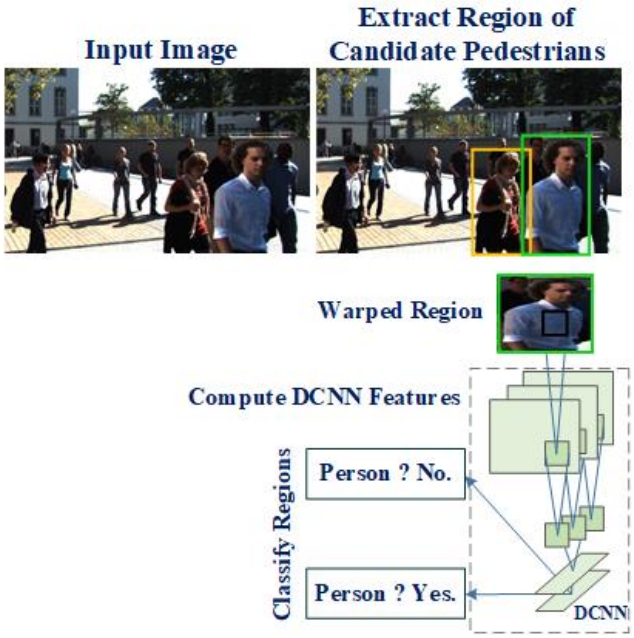

Fig. 1. Pedestrian detection using RCNN approach.

Fast RCNN Approach for Pedestrian Detection- as shown in Fig. 2, Fast RCNN [26] approach takes as input the entire image and a set of object proposals extracted from a proper OPA. For pedestrian detection, after producing feature maps in the last CNN layer, a fixed-length feature vector is extracted for each RoI (Region of Interest) and fed into a sequence of fully connected layers to estimate class and bounding-box regression of each box. The only drawback of Fast RCNN approach is that the accuracy depends on OPAs. However, Fast RCNN approach has higher computing performance than that of RCNN approach. Fast RCNN approach predicts class and bounding-box regression of each candidate pedestrian based on the results of two sibling output layers.

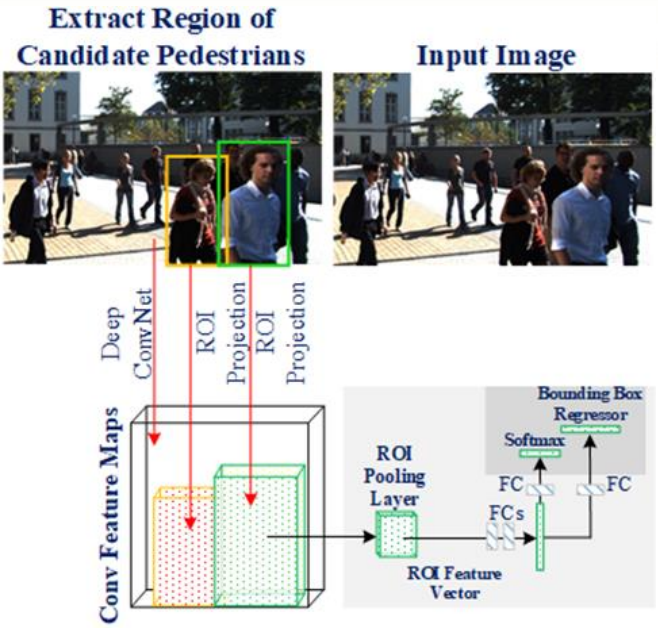

Fig. 2. Pedestrian detection using Fast RCNN approach.

Faster RCNN Approach for Pedestrian Detection- Faster RCNN [27] is composed of two modules: RPN and Fast RCNN detector. The RPN module helps the Fast RCNN module in looking for candidate pedestrians in image. RPN module slides a small network over the feature maps output by the last shared convolutional layer. At each sliding window location, RPN predicts multiple region proposals which are called anchors. Each anchor is then mapped to a lower-dimensional feature. Ultimately, this feature is fed into sibling fully connected layers to predict class and bounding-box regression of each proposed object.

As shown in Fig. 3, Faster RCNN approach utilizes a pyramid of anchors. It performs classifying and regressing bounding-boxes with reference to anchor boxes of different scales and aspect ratios. Faster RCNN differs from RCNN in a sense that it is an anchor-based method and utilizes a deep fully convolutional network to extract RoI, whereas Fast RCNN does so by an appropriate OPA.

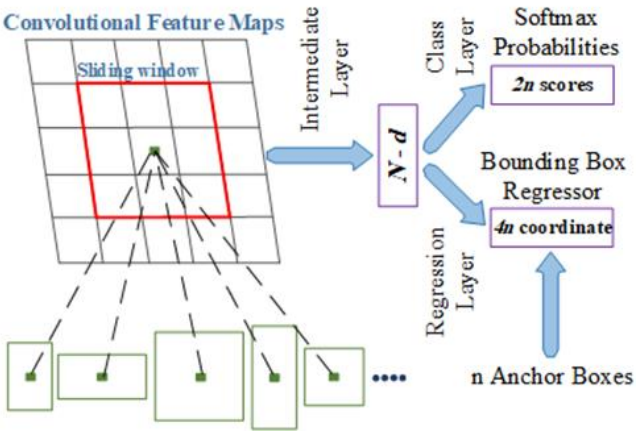

Fig. 3. Pedestrian detection using Faster RCNN approach.

C. SSD Approach for Pedestrian Detection

SSD approach predicts category scores and box offsets for a constant set of default bounding-boxes using small kernels applied to different feature maps. This training approach generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. It incorporates approximation from different feature maps to manage objects of various sizes. In other words, SSD produces predictions of different scales from feature maps of different scales, and definitely isolates predictions by aspect ratio to achieve high detection accuracy.

SSD approach adds various feature layers to the end of a basic network to predict the offsets of default boxes of different scales and aspect ratios, and their associated confidences. In fact, the main characteristic of SSD approach is to use of multi-scale convolutional bounding-box outputs attached to multiple feature maps at the top of the network.

One of the object categories detected in SSD approach is 'people' and can be considered as 'pedestrian'. However, pedestrian detection based on the architecture illustrated in Fig. 4 has not competitive accuracy in comparison with the state-of-the-art methods in pedestrian detection. To detect pedestrian, we can change the number of categories from 21 to 2.

One category being pedestrian and the other being background.

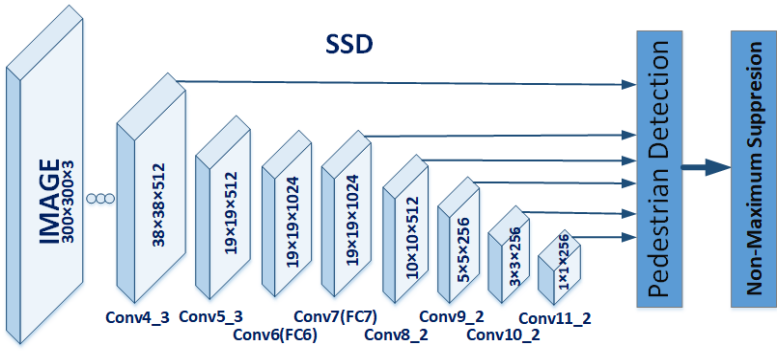

Fig. 4. Pedestrian detection using SSD network architecture.

D. *Comparing Training Approaches in Pedestrian Detection*

SWCNN, SSD, and RCNN family approaches have their own similarities and differences in pedestrian detection as follows:

- In RCNN, we have a smaller set of image patches compared to the SWCNN. In pedestrian detection based on RCNN, the image region cropped with a tight bounding-box is utilized as the input of CNN while the class label of objects within the bounding-box is estimated using a classifier. Although computational cost in RCNN is lighter than that of SWCNN, the recall of RCNN drops significantly with increasing Intersection-over-Union (IoU).
- In RCNN, each detected object is classified using Non-deep methods such as linear SVM to enhance the classification and also reduce pedestrian localization error. In contrast, there is no Non-deep classifier in Fast RCNN, Faster RCNN, SSD, and SWCNN.
- The Faster RCNN is composed of two modules: RPN, and Fast RCNN detector. The main difference of Faster RCNN and Fast RCNN is that Faster RCNN employs a deep fully convolutional network to extract RoIs, whereas Fast RCNN does so by an appropriate OPA.
- RCNN and Fast RCNN both utilize an OPA to propose candidate objects, whereas SWCNN, Faster RCNN, and SSD employ sliding windows, anchor boxes, and default boxes respectively.
- RCNN is fed by a set of proposed objects as input, whereas Fast RCNN requires whole image and a set of object proposals. Faster RCNN and SSD only need an entire image as input. In fact, Faster RCNN and SSD entirely omit proposal generation and enclose all computation in a single network.
- RCNN, Fast RCNN, Faster RCNN, and SSD approaches have a smaller set of image patches compared to the SWCNN approach.
- SSD and RCNN family perform much better than SWCNN in speed, accuracy, and bounding-box regression.
- Faster RCNN and SSD are the fastest approaches and SWCNN is the slowest in object detection.
- Similar to Faster RCNN, SSD is easy to train and relatively simpler approach compared to RCNN and Fast RCNN which require object proposals.
- The main differences between SSD and RCNN family approaches is that SSD utilizes multiple layers to detect object, whereas RCNN family approaches only utilizes the last CNN layer.

Most of the pedestrian detection algorithms based on DCNN employ RCNN approach which is a slow and multi-stage solution with costly training. The improved RCNN approaches, Fast RCNN and Faster RCNN restore the drawbacks of RCNN, share computation to speed up this approach, and enhance precision in object detection. In other words, employing modified RCNN, especially Fast RCNN and Faster RCNN approaches can perform better than the RCNN approach from viewpoints of performance computing.

III. CATEGORIZATION OF DCNN-BASED PEDESTRIAN DETECTION

Recently, a variety of pedestrian detection algorithms based on DCNN have been proposed, which can be categorized into the following five proposed:

Category 1: Algorithms based on Pre-Trained Deep Models without Amending Architecture- some DCNN-based pedestrian detection algorithms utilize pre-trained deep model without reforming network architecture. This category of algorithms only fine-tunes model parameters to extract feature maps of the last convolutional layer for pedestrian detection. In fact, no modifications are performed on DCNN architecture to improve detection accuracy. SCF+AlexNet [28] and DeepParts [5] can be considered in this category because they make use of famous deep model, Alex-Net [38] and GoogLeNet [39], respectively without modifying model.

Category 2: Algorithms based on Small CNN- in some DCNN-based pedestrian detection algorithms, CNN is only employed to extract low and mid-level features of input images. Therefore, a small number of convolutional layers exist in deep network. SDN [19] is an example of such algorithms as it utilizes small number of convolutional layers to extract low and mid-

level features to feeds them to SRBM (Switchable Restricted Boltzmann Machine). In fact, SRBM plays a crucial role in SDN, while CNN is only a small part of this deep model.

Category 3: Algorithms with Similar Architecture to the Prominent Deep Models- they modify the architecture of one of the prominent DCNNs in order to improve detection performance. As an example, TA-CNN [21] which removes one of the convolutional layers of Alex-Net, has fewer parameters at all residual layers.

Category 4: Algorithms based on Utilizing Fully Connected Parallel Layers in DCNNs- some DCNN-based algorithms in pedestrian detection, consider several fully connected parallel layers to extract and process various features of regions within the input image. For instance, PD-Sharing Features method [9], after utilizing a fully connected layer for each scale of input image, parallelizes them to perform a multi-scale model.

Category 5: Algorithms based on Incorporating Non-Deep and Deep models- several methods based on DCNN, incorporate non-deep and deep models to improve the performance of pedestrian detector. As an example, Cascading CNN and Non-Deep method [29] cascades non-deep state-of-the-art pedestrian detectors with a pre-trained CNN model such as VGG-Net [40]. This technique improves the performance of non-deep model such as ACF [41], LDCF [42] and Spatial Pooling + [43] detectors, and generalizes well to a variety of feature maps.

IV. DCNN Models in Pedestrian Detection

There are different DCNN models utilized in object detection: Alex-Net [38], GoogLeNet [39], VGG-Net [40], ZF-Net [44], SPP-Net [45], Overfeat [23], and Res-Net [46]. These deep models can be utilized in pedestrian detection to extract feature maps.

Alex-Net is one of the most remarkable deep models for efficient feature representation. This DCNN has 60 million parameters. it contains eight layers with weights, the first five is convolutional layers and the last three are fully connected layers. Although ZF-Net structurally bears similarities to Alex-Net, it has different stride and kernel sizes in the first and third convolutional layers.

SPP-Net and Overfeat are not independent models. In fact, they are two separate innovations which can be applied in DCNNs. SPP-Net is a pooling strategy in DCNNs to produce a fixed-length representation regardless of image size. whereas Overfeat is a multi-scale and sliding window approach in DCNNs for object classification and detection.

In VGG-Net, depth of network is expanded using an architecture with very small kernels: 3×3 (the smallest size to catch the notion of left/right, up/down, and center). It achieves a considerable degree of refinement by pushing the depth to 16-19 weight layers.

GoogLeNet is proposed in ILSVRC 2014. It is a very prominent DCNN for object classification and detection. in GoogLeNet architecture, by switching from fully connected to sparsely connected architectures, both depth and width are enlarged while computing cost is kept constant. GoogLeNet consists of several inception networks to cover the optimal local sparse structure of a convolutional network. Inception network is useful only at higher layers and we should keep the lower layers in traditional convolutional fashion.

Deep Residual Network, ranked first in the ILSVRC 2015 classification task, with a depth of up to 152 layers (8× deeper than VGG-Net) has lower complexity. Res-Net reformulates the layers as learning residual functions with reference to the layer inputs, as opposed to learning unreferenced functions. The 50/101/152-layer Res-Nets are more precise than the 34-layer ones by notable margins.

V. Analyzing Pedestrian Detection Algorithms based on DCNN Models

Based on the published literatures, we can analyze deep pedestrian detection algorithms from training approach, categorization, and DCNN model points of view.

Training Approach: Although pedestrian detection algorithms have been employing RCNN approach for training, the modified Faster RCNN is expected to become more prevalent in the future pedestrian detection algorithms. it not only is the fastest approach in pedestrian detection but it also is more accurate in both classification and bounding-box regression.

Categorization: From categorization point of view, although methods based on incorporating non-deep and deep models have achieved best results in pedestrian detection, taking efficiently advantages of feature maps extracted from different convolutional layers in DCNN models can reach high performance in pedestrian detection without using non-deep models. In short, we can infer that the future algorithms in pedestrian detection will focus on DCNN models much more than the current time because of efficiently disclosing and exploiting power of DCNNs in pedestrian detection.

DCNN Models: From DCNN models point of view, VGG-Net model is more popular than the other DCNN models in pedestrian detection. Researches demonstrate that GoogLeNet is more effective than VGG-Net in object classification but VGG-Net performs detecting objects precisely. Recently, a new deep model, Res-Net is proposed for object classification and detection. Res-Net is the best model from both classification and detection points of view. therefore, we can conclude that pre-trained Res-Net and VGG-Net will be employed much more often in the future DCNN-based pedestrian detection algorithms.

In summary, from classification and bounding-box regression points of view, methods based on improved Faster RCNN approach which makes use of VGG-Net and Res-Net models, detect pedestrian more accurately in compared to the methods based on other approaches.

In Table I we compare the Miss Rate of a number of recent DCNN-based pedestrian detection algorithms: CompAct-Deep [20], DeepParts [5], Cascading CNN and Non-Deep [29], TA-CNN [21], SCF+AlexNet [28], PD-Sharing Feature Map [9], and SDN [19]. As shown,

algorithms based on RCNN family approaches are more efficient than ones based on SWCNN. On the other hands, VGG-Net performs better than Alex-Net and GoogLeNet in pedestrian detection. Moreover, CompAct-Deep which yields best miss rate, picks VGG-Net rather than GoogLeNet to extract feature maps of convolutional layers.

TABLE I. COMPARISON OF DCNN-BASED PEDESTRIAN DETECTION ALGORITHMS

Algorithm	Category	DCNN Model	Training Approach	Miss Rate
CompAct-Deep [20]	5	VGG-Net	RCNN	11.75%
CompAct-Deep [20]	5	Alex-Net	RCNN	14.96%
DeepParts [5]	1	GoogLeNet	RCNN	11.89%
Cascading CNN and Non-Deep [29]	5	VGG-Net	RCNN	16.66%
TA-CNN [21]	3	Alex-Net	SWCNN	20.86%
SCF+AlexNet [28]	1	Alex-Net	RCNN	23.32%
PD-Sharing Feature Map [9]	4	Alex-Net	SWCNN	33.75%
SDN [19]	2	SRBM	SWCNN	37.87%

VI. PROPOSED APPROACH FOR DEEP PEDESTRIAN DETECTION BASED ON DCNN

We propose a parallel DCNN which effectively takes advantage of feature maps extracted from convolutional layers of DCNN models.

the proposed method first employs Faster RCNN approach to extract candidate pedestrian, then it employs six parallel DCNNs to estimate different body parts of candidate pedestrian, and ultimately merges the estimations resulted from six DCNNs to perform final classification and bounding-box regression.

As shown in Fig. 5, the proposed method employs VGG-Net, an effective DCNN model, to extract feature maps of candidate pedestrian based on Faster RCNN approach. To predict different body parts, it merges the estimations resulted from six deep models to perform ultimate classification and bounding-box regression.

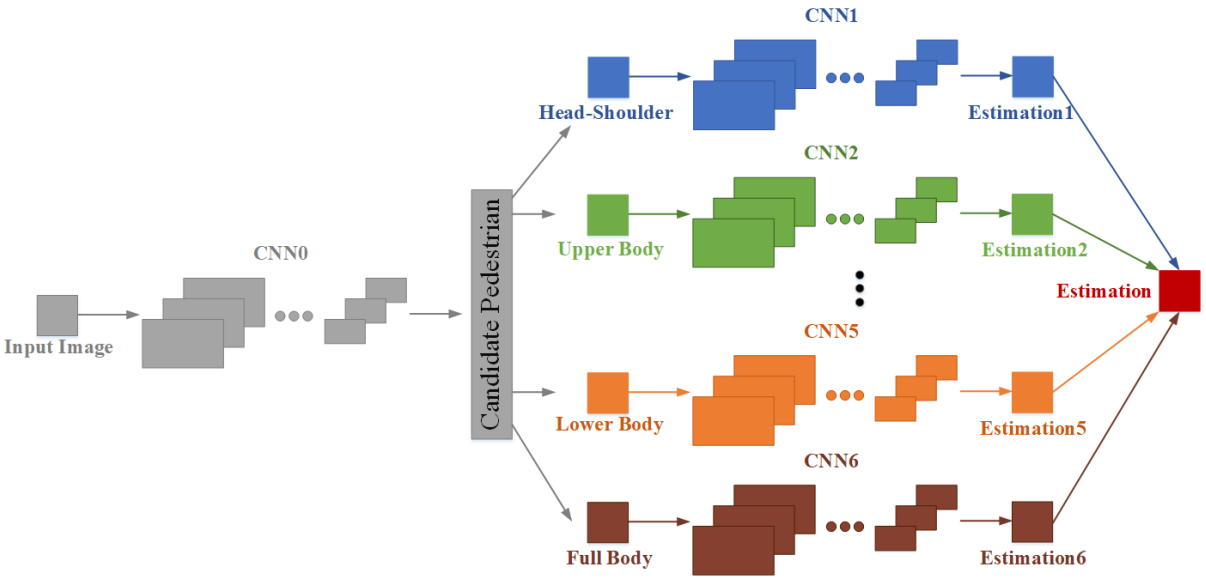

Fig. 5. Architecture of the proposed deep pedestrian detection method based on parallel DCNNs to extract feature maps of different body parts.

The proposed approach which can be considered as a principal model in the future trends of DCNN based deep pedestrian detection algorithms, includes the following stages:

Stage 1: Extracting feature maps of conv5_3 within the first model (VGG-Net).

Stage 2: Extracting a set of RoIs as candidate pedestrian using RPN.

Stage 3: Feeding RoIs to six parallel fine-tuned DCNNs.

Stage 4: Estimating simultaneously six body parts of candidate pedestrian (Full Body, Head-Shoulder, Upper Body, Lower Body, Right Body, and Left Body).

Stage 5: Feeding the scores of six parallel DCNNs to the fully connected layer.

Stage 6: Extracting the scores for each RoI from the output layer.

Stage 7: Ranking scores of detected pedestrians and employing NMS to get the final bounding-boxes as detected pedestrians.

We manage occlusion by estimating deep feature

maps of the six body parts of any candidate pedestrian. By overcoming occlusion, we achieve low miss rate in occluded pedestrian detection. On the other hand, by extracting and processing feature maps of the six parts of each RoI, we remove RoIs with low scores to reduce false positive rate. As a result, the proposed method simultaneously reduces both miss rate and false positive rate.

VII. EXPERIMENTS

Our experiments are performed on VGG-Net for estimating candidate pedestrian and Alex-Net, VGG-Net, and Res-Net for estimating six body parts of candidate pedestrian. We train the proposed method by Caltech10x and test by reasonable subset of Caltech1x pedestrian Datasets. The Reasonable subset includes pedestrian with larger than 49 pixels in height and at least 65 percent visible body parts. It is considered as a representative evaluation on all pedestrians.

A. Extracting Candidate Pedestrians based on RPN

Similar to Faster RCNN, the proposed method employs RPN to locate candidate pedestrians. RPN is an impressive network to find candidate pedestrians. After determining region of candidate pedestrians, we divide the proposed regions into six parts: Full Body, Head-Shoulder, Upper Body, Lower Body, Right Body, and Left Body.

B. Training Proposed Architecture to Detect Pedestrian

Training Data- Trainings on Caltech-USA Pedestrian Dataset [43] achieve better results than trainings on other datasets. It is the largest pedestrian benchmark and covers almost 10 hours of 640×480 30Hz video taken from a car driving through regular traffic in an urban environment. In our experiments, we utilize Caltech-USA dataset to train and test the proposed method. Due to the large number of Alex-Net and VGG-Net parameters, the size of training dataset is tremendously important for DCNNs and they require more training data to fine-tune these deep models. So we feed additional training data using Caltech10x. in fact, instead of Caltech1x which contains every 30th image in video, we utilize Caltech10x which contains every 3th image in video.

Training DCNN to find candidate pedestrian- At the first stage for training proposed architecture, we train the first DCNN to extract candidate pedestrians. To train DCNN to find candidate pedestrian, we create a mini batch with N=1 image. Each mini-batch is made up of 32 RoIs chosen randomly from only one image. We take equal or fewer than 16 of the RoIs from object proposals that have IoU with a ground-truth bounding-box of at least 0.70. Furthermore, we take equal or fewer than 16 of the RoIs from object proposals that have IoU with a ground-truth bounding-box of less than 0.30. In fact, we ignore the RoIs that have IoU between 0.30 and 0.70 to fine-tune Candidate Pedestrian Extractor Network (CPEN). We should note that if the number of RoIs is very large, then the most of them are negatives. This caused an imbalance between the positive and negative training instances. In this case, instead of considering all the negatives instances in training, we sort them using the highest confidence loss for each RoI and then pick the ones with highest ranking so that the ratio between the negatives and positives instances is at most 3:1. In our experiments, the time required to train this part of network is approximately 25 hours for 100 epochs.

Training DCNNs to Estimate Different Parts of Pedestrian- At the second stage for training proposed architecture, we train the Parallel DCNNs (PDCNNs) to estimate body parts of candidate pedestrians. Training PDCNNs is similar to training CPEN but with different. To train PDCNNs, we divide the grand-truth of CPEN into six parts (Full Body, Head-Shoulder, Upper Body, Lower Body, Right Body, and Left Body) and consider each of the parts as a grand-truth for training corresponding DCNN in PDCNNs. In our experiments, the time required to train this part of network is approximately 35 hours for 100 epochs.

C. Experimental Results

In our experiments, we utilize VGG-Net in CPEN, and three different pre-trained deep neural networks in PDCNNs: Alex-Net, VGG-Net, and Res-Net.

As shown in Table II, after fine-tuning the proposed architecture using Caltech10x training data, it has improved miss rate by 0.6%, 0.6%, and 0.7% compared to Faster RCNN in Alex-Net, VGG-Net, and Res-Net respectively. Also, the results show miss rate improvement of 0.2%, 0.3%, and 0.4% over SSD in Alex-Net, VGG-Net, and Res-Net respectively.

Table III compares Miss Rate of the proposed architecture with the previous ones after fine-tuning on Caltech10x training data. The results demonstrate that the proposed method detects pedestrian more accurately.

TABLE II. LOG-AVERAGE MISS RATE (%) ON CALTECH-USA TEST REASONABLE SUBSET AFTER FINE-TUNING ON CALTECH10X.

Approach \ Model	Alex-Net	VGG-Net	Res-Net
Faster RCNN [27]	28.8%	20.2%	19.8%
SSD [25]	28.4%	19.9%	19.5%
Proposed approach	28.2%	19.6%	19.1%

TABLE III. LOG-AVERAGE MISS RATE (%) ON CALTECH-USA TEST REASONABLE SUBSET.

Deep Algorithms	Multi-SDP [18]	Joint-Deep [47]	SDN [19]	TA-CNN [21]	Proposed approach
Miss Rate	45.4%	39.3%	37.9%	20.9	19.1%

Fig. 6, Fig. 7, and Fig. 8 illustrate the results of the proposed approach, SSD, and Faster RCNN approaches respectively, all using VGG-Net. Our proposed approach detects pedestrians with a greater accuracy.

Fig. 6. Pedestrian detection results using the proposed method based on parallel DCNNs.

Fig. 7. Pedestrian detection results using SSD approach.

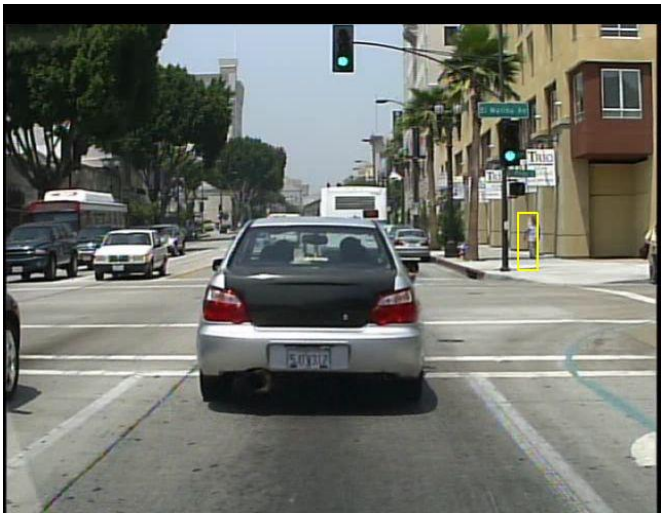

Fig. 8. Pedestrian detection results using Faster RCNN approach.

VIII. Conclusions

In this paper, after reviewing a number of state-of-the-art DCNN-based pedestrian detection algorithms, and analyzing algorithms from viewpoints of training approach, categorization, and deep model, we have proposed a new method based on parallel DCNNs and Faster RCNN approach for pedestrian detection algorithms.

As shown, VGG-Net detects objects more accurately than any other deep models such as Alex-Net, ZF-Net, SPP-Net, Overfeat, and GoogLeNet. Compared to VGG-Net, Res-Net is an enormous network which has not been employed very often in pedestrian detection. Nonetheless, one can expect that both Res-Net and VGG-Net models will be utilized in the future pedestrian detection algorithms based on DCNN.

The main training approaches for DCNN-based pedestrian detection consist of SWCNNs, SSD, and RCNNs family approaches. There are two strategies for SWCNN approach: the first is to keep DCNN unchanged and the other is to keep the input unchanged. The first strategy has some limits but it is faster than the second one. RCNN-based family approaches include RCNN, Fast RCNN, and Faster RCNN. Most of the recent pedestrian detection algorithms perform training based on this family of approaches. Similar to Faster RCNN approach, SSD is simple relative to approaches that require object proposals and also easy to train for pedestrian detection. It is expected that Faster RCNN and SSD will be more prevalent approaches in the future DCNN-based pedestrian detection algorithms because they are faster and more accurate than the other training approaches in pedestrian detection.

We categorized DCNN-based algorithms into five different categories. Although recent deep pedestrian algorithms which are based on category 5 (algorithms based on incorporating non-deep and deep models) have great performance, it is expected that the future DCNN-based pedestrian algorithms will be mainly based on categories 3 (algorithms based on similar architecture to the prominent deep models) and 4 (algorithms based on utilizing parallel fully connected in deep models). Because taking efficiently advantages of feature maps extracted from DCNN models can result in high performance for pedestrian detection without using non-deep models.

We proposed a new method based on parallel DCNNs to detect pedestrian. The proposed method includes two stages of training: Candidate Pedestrian Extractor Network (CPEN) training to extract candidate pedestrians, and secondly Parallel DCNNs (PDCNNs) training to estimate body parts of candidate pedestrians.

The proposed method exploits the extracted feature maps of DCNN to handle occlusion which consequently results in both smaller miss rate and smaller false positive rate in pedestrian detection.

Acknowledgment

Authors would like to acknowledge Iran Telecommunication Research Center, for supports throughout this research.

References

[1] D. Lowe, "Distinctive image features from scale-invariant key-points", in *IJCV*, 60(2): 91-110, 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", in *Proceedings of the CVPR*, 2005

[3] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance", in *IJCV*, 63(2): 153-161, 2005.

[4] P. Felzenszwalb, R. B. Grishick, D. McAllister, and Ramanan. (2010). Object detection with discriminatively trained part based models. *IEEE Trans. Pattern Analysis and Machine Intelligence. vol.99*. no. PrePrints.

[5] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning strong parts for pedestrian detection", in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[6] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila, "Multi-cue pedestrian classification with partial occlusion handling", in *CVPR*, 2010.

[7] T. Gao, B. Packer, and D. Koller, "A segmentation-aware object detection model with occlusion handling", in *CVPR*, 2011.

[8] C. Wang, X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling", in *CVPR*, 2009.

[9] X. Jiang, Y. Pang, X. Li, J. Pan. (2016). Speed up deep neural network based pedestrian detection by sharing features across multi-scale models. *Elsevier. Neurocomputing*. pp. 163-170.

[10] J. Yu, H. Sugano, R. Miyamoto, T. Onoye, "GPU Implementation of Efficient Pedestrian Detection Based on MCMC", in *SCIS & ISIS*, (2010).

[11] C. Cai, J. Gao, B. Minjie, P. Zhang, H. Gao. (2015). Fast Pedestrian Detection with Adaboost Algorithm Using GPU. *International Journal of Database Theory and Application. 8(6).* pp. 125-132

[12] R. Benenson, M. mathias, R. Timofte, L. Gool, "Pedestrian detection at 100 frames per second", in *CVPR*, (2012).

[13] T. Machida, T. Naito, "GPU & CPU Cooperative Accelerated Pedestrian and Vehicle Detection", in *Proceeding of IEEE ICCV Workshops*, (2011).

[14] M. Trompouki, L. Kosmidis, N. Navarro, "An Open Benchmark Implementation for Multi-CPU Multi-GPU Pedestrian Detection in Automative Systems", in *IEEE/ACM Conference on Computer-Aided Design (ICCAD)*, (2017).

[15] R. Benenson, M. Mathias, T. Tuytelaars, L. Van Gool, "Seeking the strongest rigid detector," in *CVPR*, 2013.

[16] S. Zhang, C. Bauchhage, A.B. Cremers, "Informed haar-like features improve pedestrian detection", in *CVPR*, 2014.

[17] R. Benenson, M. Omran, J. Hosang, B. Schiele, "Ten years of pedestrian detection, what have we learned", in *ECCV, CVRSUAD workshop*, 2014.

[18] X. Zeng, W. Ouyang, X. Wang, "Multi-stage contextual learning pedestrian detection", in *ICCV*, 2013.

[19] P. Luo, Y. Tian, X. Wang, X. Tang, "Switchable deep network for pedestrian detection," in *CVPR*, 2014.

[20] Zhaowei Cai, Mohammad Saberian, and Nuno Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection", in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[21] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Pedestrian detection aided by deep learning semantic tasks", in *IEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015

[22] C. Szegedy, A. Toshev, D. Erhan, "Deep neural networks for object detection", in *Proceedings of the NIPS*, 2013.

[23] P. Sermanet, D. Eigen, X. Zhang, et al., "Overfeat: integrated recognition, localization and detection using convolutional networks", in *Proceedings of the ICLR*, 2014.

[24] R. Girshick, J. Donahue, T. Darrell, et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", in *Proceedings of the CVPR*, 2014.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, "SSD: Single Shot MultiBox Detector", in *ECCV*, pp:21-37, (2016).

[26] R. Girshick, "Fast R-CNN", in *Proceedings of the ICCV*, 2015.

[27] S. Ren, K. He, R. Girshick, et al, "Faster R-CNN: towards real-time object detection with region proposal networks", in *Proceedings of the NIPS*, 2015.

[28] Jan Hosang, Mohamed Omran, Rodrigo Benenson, and Bernt Schiele, "Taking a deeper look at pedestrians," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015

[29] D. Riberio, J. C. Nascimento, A. Bernardino, G. Carneiro. (2017). Improving the performance of pedestrian detectors using convolutional learning. *Elsevier. Pattern Recognition*, pp. 641-649.

[30] P. Dollar, C. Wojek, B. Schiele, P. Perona, "Pedestrian detection: A benchmark", in *CVPR*, 2009.

[31] A. Geiger, P. Lenz, R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite", in *Conference on Computer Vision and Pattern Recognition, CVPR*, 2012

[32] A. Ess, B. Leibe, K. Schindler, L. Van Gool, "A mobile vision system for robust multi-person tracking", *in CVPR, IEEE Press*, June, 2008

[33] B. Alexe, T. Deselaers, V. Ferrari. (2012). Measuring the objectness of image windows. *Pattern Anal. Mach. Intell. IEEE Trans*. 34 (11). pp. 2189-2202.

[34] J.R.R Uijlings, K.E.A van de Sande, T. Gevers, et al., "Selective search for object recognition", *Int. J. Comput. Vis*. 104 (2) (2013) 154-171.

[35] I. Endres, D. Hoiem, "Category independent object proposals", in: "*Proceedings of the ECCV*, 2010.

[36] M.M. Cheng, Z. Zhang, W.Y. Lin, et al., "BING: binarized normed gradients for objectness estimation at 300fps", in *Proceedings of the CVPR*, 2014

[37] C.L. Zitnick, P. Dollar, "Edge boxes: locating object proposals from edges", in *Proceedings of the ECCV*, 2014.

[38] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *proceedings of the NIPS*, 2012.

[39] C. Szegedy, W. Liu, Y. Jia, et al., "Going deeper with convolutions", in *Proceedings of the CVPR*, 2015

[40] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in *Proceedings of the ICLR*, 2015.

[41] P. Dollar, R. Apple, S. Perona, "Fast feature pyramids for object detection," in *PAMI*, 2014.

[42] W. Nam, P. Dollar, J.H. Han, "Local decorrelation for improved pedestrian detection", in *NIPS*, 2014.

[43] S. Paisitkriangkrai, C. Shen, A. Van den Hengel, "Strengthening the effectiveness of pedestrian detection with spacially pooled features", in *ECCV*, 2014.

[44] M. D. Zeiler, R. Fergus, "Visualizing and understanding convolutional neural networks", in *Proceedings of the ECCV*, 2014.

[45] K. He, X. Zhang, S. Ren, et al., "Spatial pyramid pooling in deep convolutional networks for visual recognition", in *Proceedings of the ECCV*, 2014.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceeding of CVPR*, 2016.

[47] W. Ouyang, X. Wang, "Joint deep learning for pedestrian detection", in ICCV, 2013

AUTHORS' INFORMATION

Mahmoud Saeidi received his B.Sc. in Electrical Engineering from K. N. Toosi University, Tehran, Iran, in 2000 and his M.Sc. degree in Electrical Engineering from Amirkabir University, Tehran, Iran in 2003.

He is currently a Ph.D. student in K. N. Toosi University of Technology. His current research interests include Deep Learning and Pedestrian Detection.

Ali Ahmadi Received his B.Sc. in Electrical Engineering from Amir Kabir University, Tehran, Iran, in 1991 and his M.Sc. and Ph.D. degree in Artificial Intelligence and Soft Computing from Osaka Prefecture University, Japan in 2001 and 2004, respectively.

He worked as a researcher in Research Center for Nano devices and Systems in Hiroshima University, Japan during 2004–2007. He has been working as assistant professor at K.N.Toosi University of Technology Tehran, Iran, since 2007. His research interests include Distributed Intelligent Systems, Human-Computer Interaction, Computational Intelligence, Adaptive and Interactive Learning Models, Virtual Reality and Artificial Life.