Two Parallel Deep Convolutional Neural Networks for Pedestrian Detection

Bo-Yao Lin  
Institute of Information Science  
Academia Sinica, Taipei, Taiwan  
Email: boyaolin@iis.sinica.edu.tw

Chu-Song Chen  
Institute of Information Science & Research Center for Information Technology Innovation  
Academia Sinica, Taipei, Taiwan  
Email: song@iis.sinica.edu.tw

Abstract—Pedestrian detection attracts lots of attentions in the field of computer vision in recent years. It is difficult to handle data imbalance between positive and negative examples and easy-to-confused negative samples for pedestrian detection when training a single deep convolutional neural network (CNN) model. In this paper, we present a deep learning approach that combines two parallel deep CNN models for pedestrian detection. We propose using two deep CNNs, and each of which is capable of solving a particular mission-oriented task to form parallel classification models. Then, the models are integrated to build a more robust pedestrian detector. Experimental results on the Caltech dataset demonstrate the effectiveness of our approach for pedestrian detection compared to other state-of-the-art deep CNN methods.

I. INTRODUCTION

Pedestrian detection based on images is an active research area and become more and more popular within the last decade. A robust pedestrian detector is a key component applicable to the fields such as automotive safety [1], robotics [2], and visual surveillance [3]. Pedestrian detection problem is regarded as a canonical object detection task, where differentiating human bodies from backgrounds is its main goal. Detecting humans in various scenes is challenging on the grounds that humans may have various postures with partial occlusions. Furthermore, shapes of several human-like objects (such as postboxes and traffic lights) on the street may also be confused with the human bodies due to large ambiguities between these two types of objects, making the problem even challenging.

Deep convolutional neural networks (CNN) have received considerable attentions in recent years, and have shown their effectiveness for objection recognition recently. Deep CNNs have the ability to learn discriminative feature representations from raw RGB pixels in an end-to-end manner, and they unify the feature-extraction and classification in a single learning model. Deep CNNs have been widely used in image classification [4] [5], object detection and localization [6] [7], and segmentation [8]. Several deep architectures have been proposed for general object recognition tasks, eg., AlexNet [5], GoogLeNet [9], and VGG [10]. GoogLeNet has a great success on both the image-classification and object-detection tasks in the recent ILSVRC2014, where there are 1.2 million images of 1000 classes [4].

Due to the promising results of general deep CNN models, a beneficial method for training deep CNN is to adopt the parameters learned from the general image classification problem, which are pre-trained on Imagenet, as the initial weights. Then, it fine-tunes the weights for the transfer-learning purpose to solve the problems in other related domains. Such a parameter-transfer-learning scheme has achieved advantageous effects in several object detection tasks, such as R-CNN [6]. Hence, in this paper, we follow this transfer-learning principle and pre-train on Imagenet to learn a pedestrian detector. In [11], it provides a fine view of architectures and parameters to implement pedestrian detection on deep CNN models through a wide range of experiments. However, based on our observation, direct parameter-transferring or fine-tuning usually cannot solve the problem well due to the discrepancy between the pedestrian detection and classification problem. First, pedestrian detection is a 2 classes problem including pedestrian and background, where the background class is an universal concept consisting of unlimited kinds of objects or scenes. It is hard to reflect the variations of the backgrounds from a general image classification model that learns from specific 1000 object classes. Furthermore, the imbalanced problem resulting from the background training data (that are far more than the pedestrian training data) also make the fine-tuned network ineffective.

To address this problem, we use GoogLeNet in a different organization in this paper. An advantageous characteristic of GoogLeNet is that it is a flexible deep learning architecture consisting of the early layers (for learning early representations), middle layers (for deep feature extraction), and final layers (for integration and classification). In particular, “inception” is a repetitive and re-useable structure, which is employed to construct the middle layers, where the number of inceptions can be alternated to fit different task goals. In this work, we suggest to use a GoogLeNet with fewer inceptions, which can largely reduce the parameters which need to be learned through the training stage and show that it is more effective for pedestrian detection. Besides, due to above observation, we found that a single deep CNN model learned from
the imbalance training data is hard to differentiate the critical negative examples from the positive example of pedestrians. To overcome this difficulty, we introduce another deep CNN with different missions assigned and propose to combine these deep CNNs, and then mix the final-layer classification results of these models to form a pedestrian detector. We call the approach Two Parallel Deep Convolutional Neural Networks (TPDCNN). Finally, we validate its performance on Caltech dataset [12], which is a representative benchmark publicly available for pedestrian detection.

II. RELATED WORK

In this section, we review several methods designed for pedestrian detection. Existing methods for pedestrian detection can be divided into two categories: handcrafted-features and deep-learning approaches. We review these methods from these two distinct directions. Handcrafted features, eg., Haar-like features [13], SIFT [14], HOG [15], HOG-LBP [16], have been widely employed for pedestrian detection. To handle more complex articulations of human parts, several deformation models [17] [18] [19] are introduced. These deformable part models (DPM) learn a mixture of local templates for the body parts. Classifiers such as SVM [15], boosting classifiers [20], random forests [21] are then used to determine whether a pedestrian is detected. Integral channel features [20] have become a popular method to extract efficient features for pedestrian detection. Dollar et al. propose Aggregated Channel Features (ACF) [22] that comprise gradient histogram, gradients, and LUV. Then, they learn the classifiers in a boosting manner. Ke [23] et al. use convolutional network architecture with orthogonal PCA filters to improve ACF. Zhang et al. propose Checkerboards [24] that use filtered channel features with HOG+LUV as low-level features. They find that the number of filters appears to be the most important variable, and checkerboard-like patterns or purely random filters can achieve good performance. They also add optical-flow features to further enhance the performance.

Deep-learning methods perform end-to-end learning to tackle the pedestrian detection problem. In ConvNet [25], the authors use CNN to handle the limited training data on INRIA dataset. They also test the performance on Caltech pedestrian dataset while training on INRIA. DBN-Isol [26] uses a stack of Restricted Boltzmann Machines (RBMS) to extend deformable parts models (DPMs). They design overlapping parts at multiple layers and then, verify the visibility of a part for multiple times at distinct layers. DBN-Mut [27] extends the DBN-Isol to account for person-to-person relations. JointDeep [28] is proposed to use deep networks to jointly learn the feature extraction, deformation handling, occlusion handling, and classification. Zeng et al. [29] adopt deep model that can jointly train multi-stage classifiers through several stages of back-propagation and use contextual features computed at different scales. SDN [30] uses “switchable layers” to jointly learn low level features and high level parts. Hosang et al. [11] employ CifarNet and R-CNN with AlexNet for pedestrian detection. They also discuss the performance impact of pedestrian detection on different architectures and parameters. Tian et al. introduce TA-CNN [31] where further pedestrian attributes (e.g. “carrying backpack”) and scene attributes (e.g. ‘vehicle’ and ‘tree’) are introduced to solve the confusions between the positive and hard negative samples. LFOV [32] proposes a Large-Field-of-View deep network to make classification decisions simultaneously and accurately at multiple locations. Angelova et al. [33] propose to cascade deep nets and fast features to speed up the detection process while maintaining good detection performance.
III. METHOD

In this section, we would illustrate our concept and details of the proposed TPDCNN. Before introducing our method, a brief review of GoogLeNet is given in advance. First, an image fed into the network goes through the early layers shown in Figure 1(a). In early layers, some early representations are formed through a sequence of $7 \times 7$, $1 \times 1$, and $3 \times 3$ convolutions and max pooling operators. Then, middle layers which is formed by a repetitive structure called inception is concatenated with the early layers. An inception has several parallel $1 \times 1$, $3 \times 3$, and $5 \times 5$ convolutions and max pooling summarized with a depth concatenation stage, as shown in Figure 1(b). A depth concatenation is then performed and the output is sent to the next inception. The inception serves as a basic unit module to extract deep-representation features, and 9 inceptions are used in GoogLeNet. The final layers (shown in Figure 1(c)) integrate the deep features extracted by average pooling and a fully-connected layer, and then linear classifiers are constructed for the classification. The scale of learning parameters, i.e. weight, in neural networks is a pivotal factor. It may degrade the performance of networks if the number of learning parameters is too larger with limited training data. Compared to the AlexNet model having almost 60 Mega parameters, GoogLeNet reduced the learned parameters to approximately 5 Mega with a more deeper network.

In the original GoogLeNet, it uses softmax to classify 1000 categories on ImageNet. For pedestrian detection, it only needs two classes, pedestrian and background, which is a binary classification problem. In our work, we replace the softmax layer with a sigmoid cross-entropy loss in the final linear classifier layer. Nevertheless, using the whole GoogLeNet for pedestrian detection often suffers from severe over-fitting problem, even though the networks are fine-tuned from a diverse and large-scale dataset, ImageNet. The result of utilizing GoogLeNet to train a pedestrian detector is shown in Section IV. In the following, we introduce a concise model to tackle this problem in Section III-A. Then, the joint models are depicted in Section III-B.

A. Concise model for pedestrian detection

The original GoogLeNet is a mighty model that can differentiate various objects. However, the features learned in the middle layers of the GoogLeNet would be too restrictive to be fine-tuned to a pedestrian detector. Nevertheless, it would have a better generalization ability of the features learned in some previous layers (based on the ImageNet) as they are not that restrictive to the particular object classes.

This inspired us to use a concise model for handling the pedestrian detection problem. In our architecture, the model is reduced to a concise network with fewer inceptions in the middle layer (where 2 inceptions are used here). We retained the early layers and final layers in our concise model. This makes the training process easier and faster because the overall parameters are reduced. Besides, we still make use of pre-training and initialize the weights of the early and middle layers with that learned from the ImageNet. The concise model is illustrated in the upper row of Fig. 2. Compared to GoogLeNet that achieves the log-average miss-rate (MR) of 36.66%, our concise model can considerately reduce the MR to 27.31% on Caltech test set.
B. Two Parallel Deep Convolutional Neural Networks

Inter-class correlation, where some background regions are similar to the pedestrians, is a main issue for pedestrian detection. Region proposals consisting of a large portion of background area and a small portion of pedestrian area are also critical negative samples resulting in the inter-class correlation problem, and this may incur localization error. The concise model introduced in Section III-A still suffers from the critical negative background samples that have large ambiguity and are easy to be confused with the foreground pedestrians.

We adopt a mission-oriented strategy for training the networks to deal with this difficulty. The networks trained with different goals are combined to form the TPDCNN architecture. Currently, TPDCNN has two rows (with each row a separated model) as shown in Fig. 2. Without loss of generality, it can be added with more rows for different tasks if necessary. The mission of the first deep model is to discriminate the pedestrian samples with general and easy negative background samples, while the mission of the other model is to discriminate them with critical negative examples only. We illustrate their details as follows.

Upper row of TPDCNN: The upper row of TPDCNN consists of the early layers, middle layers of 2 inceptions, and final layers, as depicted in Section III-A. To train this model, we use all of the available positive (pedestrian) samples and negative (background) samples. The ground-truth windows provided by the dataset are employed as positive samples. In [11], it shows that adding some jitters, that is a positive proposal of large overlapping area with ground truth, may degrade the detection performance. Instead, we employ horizontal mirror to augment these positive samples. To collect the negative samples, we follow the conventions in relevant studies such as [31] that an object-proposal method is used to produce multiple candidate windows at first. Then, the windows whose overlapping degrees with the ground truths are less than a threshold (here, 0.3) are chosen to be the negative samples. Both positive and negative samples are normalized into the size of $224 \times 224$ and fed into the deep CNNs. The object-proposal method and procedure of normalization on input images used in this work will be introduced in Section III-C.

Lower row of TPDCNN: We bestow a different mission on the lower row of TPDCNN. The lower row of TPDCNN focuses on a more difficult goal where harder negative samples can be separated with the positive ones in the learned feature space (but the easier negative samples could be sacrificed). As the mission assigned is more difficult, we increase one additional inception and adopt a more complex network with three inceptions in the middle layer, as shown in the lower part of Fig. 2. Due to the relatively scarce amount of positive samples, the positive samples used for training this row remain the same as that used in the upper row, but the negative samples are chosen as the ones fail to be classified correctly by the first row. Hence, the mission of the lower row is disparate to that the first row and they are trained independently.

There are several advantages of the TPDCNN model, and we are discussed as follows. For building a pedestrian detector, hard negative human-like background regions are always a main issue. Increasing the number of hard negative samples in the training stage by changing the overlapping degree threshold may improve the performance but also causes a large imbalance between the positive and negative samples. The proposed TPDCNN can handle the data imbalance problem more properly and thus get better performance. Furthermore, if the two models are dependent or coherent, combining them becomes less meaningful. The lower network has a stronger ability to differentiate such ambiguity (although would suffer from worse performance on distinguishing general negative and positive samples) because only the negative data that are ambiguous to the upper row are used in training the lower network. These two rows are thus incoherent and mutually beneficial to each other. Finally, the final score is generated by averaging the final layers of these two parallel networks.

C. Preprocessing and postprocessing

We illustrate the implementation details of preprocess- and postprocessing steps to complete the pedestrian detection task.

1) Preprocessing: To detect all of the pedestrians in an image, a possible way is to use a sliding window to test numerous potential regions within an image. However, it is exhaustive and slow because of a large amount of computations of convolution operations in deep CNN models. Many recent studies generate the object proposals (candidate regions) to avoid the exhaustive search. SelectiveSearch [34] is the most popular proposal method and widely used in object detection. Using class-specific proposals allows to reduce the number of proposal by large magnitude. In this work, we follow the approach of [31] that uses the ACF detector [22] to generate the object proposal (via a loose threshold) for pedestrian detection. The model windows we choose for pedestrian detectors is $128x64$ pixels in which pedestrians have height 100 and width 41. The height of the region proposal extracted by ACF is normalized to 224 and the aspect ratio is preserved. As the input to the network is $224 \times 224$, the empty parts occurred in the aspect-ratio resizing are compensated with the mean color used in the GoogLeNet training.

2) Postprocessing: According to the observation on the work of objection detection [6], deep neural networks are not reliable enough on localizing objects. To increase the precision of the detected region, we also adopt the bounding-box (BBox) regression method suggested in RCNN [6] and train a ridge-regression model that predicts a new detection bounding box based on the feature extracted in the convolution layer outputs. To avoid multiple responses in an adjacent local area, we run a greedy Non-Maximum Suppression (NMS) procedure [35] to suppress the large overlapping areas of detections with lower scores.

Finally, we concatenate the preprocessing (object proposal), TPDCNN, and postprocessing (BBox regression + NMS) steps to form the entire approach. To implement TPDCNN, we modify the GoogLeNet implementation [36] on the Caffe [37] platform.
B. Evaluations of TPDCNN

In this section, we display the performance of our proposed TPDCNN and the comparison to other deep learning methods for pedestrian detection. We start to show the selection of concise models from GoogLeNet, which are used to form two parallel deep CNNs. Table I shows the performance with a different number of inceptions for upper row and lower row individually. We can see that the MR is closer on upper row when the number of inceptions in middle layers is 2 or 3. However, as the lower row is assigned to deal with more hard negative samples, the more inceptions (here, 3) is advantageous to extract better features for discrimination.

Then, we show the detection quality on different deep architectures and our proposed TPDCNN in Table II. We can see that the original GoogLeNet structure is not promising in pedestrian detection compared to other structures (even when fine-tuning is performed), and it is even worse than the proposal ACF. It shows that the original deeper GoogLeNet is not a proper architecture for pedestrian detection. Our upper-row model achieves a more favorable performance than GoogLeNet. The lower-row model (trained on the positive and critical negative samples only) can achieve comparable performance of the upper-row model. Both models perform better when bounding-box regression is used. By using TPDCNN, the joint scheme boosts the performance compared with the above deep models. Similarly, introducing the bounding-box regression on TPDCNN can also achieve a more favorable result.

In the final part, we compare our approach, TPDCNN, with other pedestrian detection approaches using deep learning. These deep learning methods include ConvNet [25], DBN-Isol [26], DBN-Mut [27], MultiSDP [29], JointDeep [28], SDN [30], LFOV [32], DeepCascade, DeepCascade+ [33], R-CNN (AlexNet) [11], and TA-CNN [31]. Table III shows the results. Our TPDCNN achieves 19.57% with the bounding box regression, which performs more favorable than the other competitive deep learning methods for pedestrian detection [32, 33, 11, 31]. Figure 3 shows the overall performance on Caltech test dataset (reasonable) with all deep models.
V. CONCLUSION AND FUTURE WORK

In this paper, we present the TPDCNN approach as a robust pedestrian detector. In order to address the problems resulting from data imbalance and critical negative examples in training a pedestrian classifier, TPDCNN combines two deep CNN models that are assigned with distinct missions during the training phase. We introduce a concise structure from GoogLeNet with fewer inceptions and build our proposed models based on this structure at first. The proposed TPDCNN model is then combined with ACF object proposal and bounding box regression to enhance the efficiency and reliability of pedestrian detection. The performance is boosted as we integrate these two deep CNNs to form TPDCNN. Our approach achieves more favorable performance on Caltech dataset, a widely adopted public benchmark, than the other deep-learning pedestrian-detection approaches compared.

In the future, we plan to fortify this method by combining the two parallel networks as a whole rather than just averaging the outputs of the individual networks. Introducing more than two parallel networks may also improve this method. Also, considering dynamic features (such as optical flow) and additional pedestrian- or scene-related attributes as a multi-task learning is also potentially useful for further performance improvement.

ACKNOWLEDGMENT

This paper was supported in part under the project MOST 104-2221-E-001-023-MY2.

REFERENCES