PEDESTRIAN DETECTION BASED ON FAST R-CNN AND BATCH NORMALIZATION

Zhong-Qiu Zhao\textsuperscript{a}, Haiman Bian\textsuperscript{a}, Donghui Hu\textsuperscript{a}, Hervé Glotin\textsuperscript{b}

\textsuperscript{a} College of Computer Science and Information Engineering
Hefei University of Technology
Hefei, China, 230009

\textsuperscript{b} Systems and Information Sciences lab
LSIS CNRS & Univ. of Sud-Toulon Var - La Garde, France
\{z.zhao, hudh\}@hfut.edu.cn
bhm2164@163.com
h.glotin@gmail.com

ABSTRACT

Most of the pedestrian detection methods are based on hand-crafted features which produce low accuracy on complex scenes. With the development of deep learning method, pedestrian detection has achieved great success. In this paper, we take advantage of a convolutional neural network which is based on Fast R-CNN framework to extract robust pedestrian features for efficient and effective pedestrian detection in complicated environments. We use the EdgeBoxes algorithm to generate effective region proposals from an image, as the quality of extracted region proposals can greatly affect the detection performance. In order to reduce the training time and to improve the generalization performance, we add a batch normalization layer between the convolutional layer and the activation function layer. Experiments show that the proposed method achieves satisfactory performance on the INRIA and ETH datasets.

1 INTRODUCTION

In recent years, pedestrian detection has become an important branch of object detection and has attracted wide attention in computer vision \cite{Dalal&Tiggs, Doll\`ar et al. 2014, 2009, Viola et al. 2005, Wang et al. 2009}. In real life, it is a key problem in automotive safety, video surveillance, smart vehicles and intelligent robotics. Because of the diversity of pedestrian body pose, object occlusions, clothing, lighting change and complicated backgrounds in the video sequence or image, the pedestrian detection is still a challenging task in computer vision.

In pedestrian detection, feature extraction is an important factor to influence the performance. Many features have been proposed by researchers for pedestrian detection, such as Haar-like features \cite{Viola & Jones 2004}, Integral Channel Features (ICF) \cite{Doll\`ar et al. 2009}, Histogram of Oriented Gradients (HOG) \cite{Dalal & Triggs 2005}, Local Binary Pattern (LBP) \cite{Mu et al. 2008}, Dense SIFT \cite{Vedaldi et al. 2009} etc. And the proposed Deformable Part Based Model (DPM) \cite{Felzenszwalb et al. 2010} which is based on HOG, has made a breakthrough in pedestrian detection. However, these features are hand-craft and are considered to be low-level. They use the low-level information while the high-level information is usually very important for pedestrian detection \cite{Sermanet et al. 2013}.

Recently, with the development of deep learning techniques, deep neural networks have been successfully applied in object recognition tasks \cite{Girshick et al. 2014, He et al. 2015, Girshick 2015, Zhao et al. 2014}. Deep neural networks can achieve more excellent results due to their capability to learn discriminative features from raw pixels. Many researchers have applied deep learning techniques to pedestrian detection. \cite{Sermanet et al. 2013} proposed a two-layer convolutional model which adopts convolutional sparse coding to pre-train convolutional neural network for pedestrian detection. \cite{Chen et al. 2014} proposed a pre-trained Deep Convolutional Neural Network (DCNN) to learn features from ACF \cite{Doll\`ar et al. 2014} detector. These
features are then fed to a SVM classifier. Ouyang et al. [Ouyang & Wang (2013)] proposed a joint deep model that jointly learns four key components in pedestrian detection: feature extraction, deformation handling, occlusion handling and classifier. Li et al. [Li et al. (2015)] proposed a SAF R-CNN which takes scales of pedestrians into account in pedestrian detection. Fast R-CNN [Girshick (2015)] is one of the best detection approaches based on deep leaning. It determines regions of interest (RoI) in an image using superpixel method, and then extracts features using convolutional neural network model. The extracted features are then passed to softmax layer and bounding box regressor layer which are trained for each object, and finally the pedestrian and its position are detected.

It is well known that training a deep neural network is complicated due to the fact that internal covariate shift [Ioffe & Szegedy (2015)] can degrade the efficiency of training. Internal covariate shift is a change of distribution of the inputs. Which happens during training the feed-forward neural networks, by changing the parameters of a layer. Batch normalization (BN) [Ioffe & Szegedy (2015)] is a technique to control the distributions of feed-forward neural network activations, thereby reducing internal covariate shift. So deep neural networks trained with batch normalization can converge faster and generalize better. In this paper, we use Fast R-CNN architecture based on batch normalization (BN) to extract robust pedestrian features.

In pedestrian detection task, another important stage is locating the potential windows which may contain pedestrians. As an exhaustive and traditional method, sliding window method has two main shortcomings. First, it needs searching every possible position in an image. Second, it may produce many redundancy windows which affect the quality of detection. To improve the detection efficiency, the approaches of region proposals have been proposed to produce high quality regions, in which the most popular method is Selective Search [Uijlings et al. (2013)]. It can get high quality of regions by using image segmentation. However, its speed is very slow and the detection performance is sensitive to the image quality. And the Selective Search method of extracting candidate windows is infeasible [Chen et al. (2014)] but cannot provide precise localization of pedestrians. To produce more precise localized candidate windows, we employ EdgeBoxes [Zitnick & Dollar (2014)] method, which is a fast and effective one [Hosang et al. (2014)]. It is believed that the edge information can precisely describe the object, so the EdgeBoxes method can extract higher quality of the candidate windows than the Selective Search method. Moreover, the EdgeBoxes method actually performs better in terms of the average intersection over union (IoU) across all images in the set [Hosang et al. (2015)].

In this paper, our main contributions are summarized as follows:

1. In order to make the pedestrian detection more efficient, we use the EdgeBoxes method which can obtain low-redundancy and high quality of candidate windows.
2. We use the Fast R-CNN architecture to extract robust features for pedestrian detection. In order to reduce the training time and prevent the gradients that explode or vanish when training the Fast R-CNN, we add the batch normalization (BN) layer into the Fast R-CNN architecture.

The rest of this paper can be organized as follows. In Section 2, we introduce the related work of pedestrian detection and briefly introduce the batch normalization layer. In Section 3, we introduce our pedestrian detection approach. In Section 4, we present our experiment results on two benchmark datasets, and some analyses as well. In Section 5, we conclude our work and discuss the possible advances to our model in the future.

2 Related Work

In this section, we review the related work of two important stages of pedestrian detection, viz. extracting region proposals and extracting features, respectively. And, we also briefly introduce the batch normalization layer.

Region Proposals: In an image, the position of a pedestrian can be anywhere and its size can be arbitrary, so it is necessary to search the whole image to localize the pedestrians. Traditional methods employ sliding window method to find the possible locations on the whole image. The sliding window method can get almost all potential locations through exhaustive search, but it is computational and may produce many redundancy windows. Some region proposals methods were proposed to overcome this problem [Hosang et al. (2015)], in which the most popular method is
Selective Search \cite{Uijlings2013}. It is a combination of exhaustive search and segmentation method. Wang et al. \cite{Wang2015} proposed a region proposal fusion algorithm to fuse the Selective Search and BING \cite{Cheng2014} method. Nam et al. \cite{Nam2014} proposed LDCF method to extract regions for pedestrian detection. The EdgeBoxes method has been shown as a state-of-the-art region proposal system \cite{Zitnick2014, Hosang2014}.

**Extracting features:** Many models based on hand-crafted features have been utilized for pedestrian detection \cite{Dalal2005, Dollar2009, Wang2009, Viola2004, Mu2008}. Wang et al. \cite{Wang2009} utilized the combination of HOG and LBP to handle the partial occlusion of pedestrian. Felzenswalb et al. \cite{Felzenswalb2008} further improved the detection performance by combining the HOG with a deformable part model. Dollár et al. proposed the Integral Channel Features (ICF) \cite{Dollar2009} and Aggregated Channel Features (ACF) \cite{Dollar2014} which efficiently extract histograms and haar features. Chen et al. \cite{Chen2013} used a multi-order context representation to take advantage of co-occurrence contexts of different objects. Deep learning models can be trained end-to-end to extract robust features from the given images. Sermanet et al. \cite{Sermanet2013} proposed a convolutional neural network model which has two convolutional layers pre-trained by sparse coding. Ouyang et al. \cite{Ouyang2013} proposed a joint deep model which contains a deformation layer to model mixture poses information. Tian et al. \cite{Tian2015} jointly utilized semantic tasks to optimize pedestrian detection. More recently, some excellent works have been successfully applied in object detection \cite{Girshick2014, He2015, Girshick2015}, and the performance of pedestrian detection is largely improved \cite{Chen2014, Li2015, Wang2015}. Fast R-CNN \cite{Girshick2015} is one of the best detection approaches among them. It consists of two steps for object detection, in which the first step is to use non-deep learning methods to extract thousands of region proposals, and the second step is to use convolutional neural networks classifiers to classify categories at those locations.

**Batch Normalization:** Training a deep neural network is complicated due to the fact that changing the parameters of a layer will affect the distribution of the inputs to the succedent layers. As a result, the succedent layers are continually adapted to fit the shifting input distribution and unable to learn effectively. This phenomenon is called Internal Covariate Shift (ICS) \cite{Ioffe2015}. The ICS slows down the course of training and needs careful parameter initialization. Batch normalization (BN) \cite{Ioffe2015} is a technique to accelerate training and to improve generalization capability. Its basic idea is to standardize the internal representations inside the network to make the network converging faster and generalizing better, inspired by the way to whiten the network input to improve performance. Batch normalization layer can be applied to convolutional neural networks directly between the convolutional layer and the activation function layer, for example ReLU etc. It uses the local mean and variance computed over the minibatch $x$, and then corrects with a learned variance and bias term, namely $\gamma$ and $\beta$:

$$BN(x; \gamma, \beta) = \beta + \gamma \frac{x - E[x]}{(Var[x] + \epsilon)^{1/2}}$$

(1)

where $\epsilon$ is a small positive constant to improve numerical stability.

3 Our Approach

In this section, we will introduce the details of our approach for pedestrian detection. Firstly, we will introduce the proposal generation algorithm. Then, we will describe the pedestrian detection architecture. Finally, we will briefly introduce the whole pedestrian detection process.

3.1 Pedestrian Proposals

We utilize the EdgeBoxes as our proposal generation algorithm. The basic idea of EdgeBoxes is that it generates and scores the proposal based on the edge map of the given image.

Specifically, firstly, a structured edge detector is utilized to generate an edge map where each pixel contains a magnitude and orientation information of the edge. Secondly, the edges are grouped together by a greedy algorithm. Then, the affinity between two edge groups is computed. This is the input to the next step for the score computation. Thirdly, for a given bounding box, the score of the
Bounding box is computed based on the edge groups entirely inside the box. Finally, as the EdgeBoxes detector may generate many overlapping candidate pedestrian windows, the non-maximal suppression is used to filter out these overlapping bounding boxes by the following criterion:

\[
\text{OverlapArea} = \frac{(\text{IntersectionArea})}{(\text{UnionArea})}
\]

If two bounding boxes overlap by more than 50%, then the bounding box with the highest score is selected.

Compared with the Selective Search which is the most popular proposal method, EdgeBoxes is much faster. For EdgeBoxes, the average runtime is about 0.3 seconds per image while Selective Search is about 10 seconds per image. In the case of accuracy on the PASCAL VOC 2007 dataset, the mAP of Selective Search is 31.7% which is slightly smaller than the mAP of EdgeBoxes 31.8% Hosang et al. (2014), and on the INRIA dataset, the recall rates of Selective Search and EdgeBoxes are 23% and 93% respectively by our experiment. In brief, the EdgeBoxes can not only reduce the runtime complexity but also increase the accuracy. Therefore, we choose the EdgeBoxes as our proposal generation algorithm.

### 3.2 The Architecture for Pedestrian Detection

![Figure 1: The architecture of our proposed pedestrian detection model, where the batch normalization layer is placed between any one convolutional layer and the succedent activation function layer (such as ReLU layer, and details can be seen in Figure 2).](image)

![Figure 2: The first five convolutional layers of our proposed model. The batch normalization (BN) layer is placed between each convolutional layer (Conv) and the succedent ReLU layer. In the first two convolutional layers, the max-pooling layer is placed to follow the ReLU layer.](image)
pooling layer is a region of interest (RoI) pooling layer. The RoI pooling layer is utilized to pool the feature maps of each input object proposal into a fixed-length feature vector which is then fed into the fully connected layers. The next two layers are fully connected layers, both of which contain 4096 nodes. At the end of the architecture there are two output layers which produce two output vectors for per object proposal. Specifically, one is a softmax layer, which outputs classification scores over K object classes plus a “background” class. The other is a bounding-box regressor layer, which outputs four real-valued numbers for each of the K object classes. These $4 \times K$ values encode refined bounding boxes for each class.

3.3 Pedestrian Detection Process

The entire pedestrian detection process can be summarized as follows:

(1) Run the EdgeBoxes algorithm to generate region proposals.
(2) Input the whole image and a number of region proposals to our proposed model.
(3) Extract the features of region proposals with convolutional neural networks.
(4) Locate and verify pedestrian in each region proposal with extracted features by softmax classifier and bounding-box regressor.

The overall framework is also presented in Figure 3.

![Figure 3: The overview of our pedestrian detection system. (1) Use EdgeBoxes to generate region proposals. (2) Pass the region proposals to the pedestrian detector which is our proposed model. The detector will produce softmax scores for each proposal bounding box. (3) For each pedestrian class, use the non-maximum suppression (NMS) independently to greedily merge the overlapped proposals.](image)

4 Experiments and Analyses

In this section, we evaluate our proposed pedestrian detection method on two popular benchmark datasets, INRIA and ETH. We follow the evaluation protocols proposed in [Dollar et al., 2012]. The log average miss rate is used to summarize the detector performance. It is computed by averaging the miss rate at 9 FPPI (false positives per image) rates ($0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.64, 0.80, 1$) which are evenly spaced in the log-space ranging from $10^{-2}$ to $10^{0}$. In one test image, if a detected bounding box $BB_{dt}$ can significantly cover most area of the body window $BB_{gt}$ of the ground-truth pedestrian, the detection window is considered to properly detect the pedestrian.

$$\alpha_0 = \frac{\text{area}(BB_{dt} \cap BB_{gt})}{\text{area}(BB_{dt} \cup BB_{gt})} \geq \theta$$

where $\alpha_0$ is the area of overlap between $BB_{dt}$ and $BB_{gt}$, and $\theta$ is the threshold which is set to be 0.5. And the experiments are run on an Ubuntu 14.04 64 bit system with a single 4GB memory NVIDIA GeForce GTX 980 GPU.

4.1 INRIA Dataset

The INRIA dataset is a very popular dataset for pedestrian detection. The annotations of pedestrians are of high quality in diverse conditions and poses. The details of INRIA dataset are listed in Table 1.

Our proposed model is implemented based on the publicly Caffe platform [Jia et al., 2014] and the CaffeNet model is taken as our initial model which has been pre-trained on the ImageNet dataset. In
To use the Fast R-CNN method, we replace the last max-pooling layer of the CaffeNet with the RoI pooling layer to pool the feature maps of each region proposal into a fixed resolution, i.e. $6 \times 6$. We also replace the final fully connected layer and softmax layer with two sibling fully connected layers, i.e. softmax layer and bounding-box regressor layer.

In order to expand positive example set, we map each region proposal to the ground-truth which has a high intersection over union (IoU). If the IoU is larger than 0.5, then we will label the selected region proposal as a positive sample for pedestrian class. The rest region proposals, whose IoUs are smaller than 0.1, are regarded as negative instances. The proposed model is trained with the Stochastic Gradient Descent (SGD) method with the momentum of 0.9 and the weight decay factor of 0.0005. Since batch normalization layer is added, a large learning rate is used in our experiment. We set the initial global learning rate as 0.01 and the dropout ratio as 0.1 in the fc6 and fc7 layers. The whole process of training is 40000 iterations, the training time of our model (FRCNN+BN) is 0.094 seconds per iteration and that of Fast R-CNN is 0.153 seconds per iteration. The average time to process an image using our model is about 0.5 seconds, while the average time of the Fast R-CNN is about 0.6 seconds.

The overall experimental results are reported with the DET curves (miss rate versus FPPI) in Figure 4. We compare our proposed model with existing methods, including two classical methods VJ Viola & Jones (2004), HOG Dalal & Triggs (2005) and other state-of-the-art methods such as ConvNet Sermanet et al. (2013), ACF Dollár et al. (2014), VeryFast Rodrigo et al. (2012), FRCNN Girshick (2015), Roerei Rodrigo et al. (2013), Sketch Tokens Joseph et al. (2013) and SpatialPooling Paisitkriangkrai et al. (2014). We also give the digitized results of the miss rate at 0.1 FPPI.

From Figure 4 we can see that our method outperforms most of the state-of-the-art methods. The miss rate of our method is 12%, which has an improvement of 8% over ConvNet and an improvement of 2% over FRCNN, but a slight decrease over the SpatialPooling method.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive images</td>
<td>614</td>
<td>288</td>
</tr>
<tr>
<td>Negative images</td>
<td>1218</td>
<td>453</td>
</tr>
<tr>
<td>Annotated windows</td>
<td>1237</td>
<td>589</td>
</tr>
</tbody>
</table>

Figure 4: Comparison of miss rate versus false positives per image between different methods on the INRIA dataset. Our model is denoted as FRCNN+BN. We obtain the miss rate of 12% at 0.1 FPPI.
4.2 ETH Dataset

The ETH dataset derives from the binocular vision of pedestrian dataset. It is obtained by a pair of on-board camera and it gives the annotations of pedestrians. The set01, set02 and set03 are currently the most used three subsets. The details of these three subsets can be seen in Table 2.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>set01</th>
<th>set02</th>
<th>set03</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive images</td>
<td>999</td>
<td>446</td>
<td>354</td>
</tr>
<tr>
<td>Negative images</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Annotated windows</td>
<td>8466</td>
<td>3472</td>
<td>2225</td>
</tr>
</tbody>
</table>

In order to evaluate the generalization performance of our proposed method, we train our proposed method on the INRIA dataset, and then apply the trained model to the ETH dataset. The performance evaluation is the same as that on the INRIA dataset. The average time to process an image using our model (FRCNN+BN) is about 0.4 seconds, while the average time of the Fast R-CNN is about 0.5 seconds. We compare our model with other methods, such as ConvNet [Sermanet et al., 2013], JointDeep [Ouyang & Wang, 2013], FRCNN [Girshick, 2015], SDN [Luo et al., 2014], TA-CNN [Tian et al., 2015], VJ [Viola & Jones, 2004], HOG [Dalal & Triggs, 2005], ACF [Dollar et al., 2014], VeryFast [Rodrigo et al., 2012]. The comparison results are shown in Figure 5, from which we can see that our method outperforms VJ, HOG, VeryFast, ACF, ConvNet, FRCNN and JointDeep but underperforms SDN, SpatialPooling and TA-CNN methods.

![Figure 5: Comparison of miss rate versus false positives per image between different methods on the ETH dataset. Our method is denoted as FRCNN+BN. We obtain the miss rate of 44% at 0.1 FPPI.](image)

From Figure 5, we can also see that the performance on the ETH dataset is worse than that on the INRIA dataset. One reason may lie in that the image quality of ETH dataset is much worse than that of the INRIA dataset, and the image resolution of the ETH dataset is lower. Another reason may be that we use an external dataset to train the model for the ETH dataset.

Finally, we also give some of our detection examples in Figure 6.

5 Conclusions

In this paper, we propose a novel deep pedestrian detection method based on the Fast R-CNN framework. To reduce the training time and improve the generalization performance, we add the batch
normalization layer between the convolutional layer and activation function layer. To further improve the pedestrian detection speed and accuracy, we use the EdgeBoxes algorithm to remove the redundant windows with poor quality. The experiments show that the proposed method can achieve satisfactory performance comparable with other state-of-the-art methods on two popular pedestrian detection benchmark datasets, INRIA and ETH.

As the image quality can affect the final detection performance, using our proposed method to achieve satisfactory detection performance on low resolution image is still an open issue.

![Detection examples from INRIA dataset (a, b, c) and ETH dataset (d, e, f).](image)

Figure 6: Detection examples from INRIA dataset (a, b, c) and ETH dataset (d, e, f).

**ACKNOWLEDGMENTS**

This research was supported by the National Natural Science Foundation of China (No.61672203 & 61375047), the Program for Changjiang Scholars and Innovative Research Team in University of the Ministry of Education of China (No.IRT13059), and the Fundamental Research Funds for the Central Universities of China.
REFERENCES


Yadong Mu, Shuicheng Yan, Yi Liu, Thomas Huang, and Bingfeng Zhou. Discriminative local binary patterns for human detection in personal album. In IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, 2008.


