Journal of Autonomous Intelligence (2019) Volume 2 Issue 1.
doi:10.32629/jai.v2i3.57

Original Article

Pedestrian detection in driver assistance using SSD and PS-GAN

Kun Zheng[1], Mengfei Wei[1], Shenhui Li[1], Dong Yang[1], Xudong Liu[1*]

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

ABSTRACT

Pedestrian detection is a critical challenge in the field of general object detection, the performance of object detection has advanced with the development of deep learning. However, considerable improvement is still required for pedestrian detection, considering the differences in pedestrian wears, action, and posture. In the driver assistance system, it is necessary to further improve the intelligent pedestrian detection ability. We present a method based on the combination of SSD and GAN to improve the performance of pedestrian detection. Firstly, we assess the impact of different kinds of methods which can detect pedestrians based on SSD and optimize the detection for pedestrian characteristics. Secondly, we propose a novel network architecture, namely data synthesis PS-GAN to generate diverse pedestrian data for verifying the effectiveness of massive training data to SSD detector. Experimental results show that the proposed manners can improve the performance of pedestrian detection to some extent. At last, we use the pedestrian detector to simulate a specific application of motor vehicle assisted driving which would make the detector focus on specific pedestrians according to the velocity of the vehicle. The results establish the validity of the approach.

Keywords: *Pedestrian Detection; Driver Assistance; GAN; SSD*

ARTICLE INFO
Received: Feb 18, 2019
Accepted: Apr 2, 2019
Available online: Apr 13, 2019

***CORRESPONDING AUTHOR**
Xudong Liu, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; hicx@bjut.edu.cn;

CITATION
Kun Zheng, Mengfei Wei, Shenhui Li, Dong Yang and Xudong Liu. Pedestrian detection in driver assistance using SSD and PS-GAN. Journal of Autonomous Intelligence 2019; 2(1): 79-89. doi: 10.32629/jai.v2i3.57

1. Introduction

Pedestrian detection is a very popular topic in the field of computer vision, which has wide applications such as automatic driving, intelligent surveillance, human behavior analysis, and mobile robotics[1–5]. For now, more and more pedestrian detection systems are applied to automobiles to save people lives. In last few years, the significant role of computer vision systems is emphasized for accident prevention. The visual system must detect the front scene and warn the driver in advance if there is an unexpected situation. Absolutely, high quality speed and accurate systems that can perform detecting on all different scene is highly satisfactory, but until now, unfortunately, no such system exists. A few methods that detect body parts of people[6,7] have been elaborated. HOG (Histogram of Oriented Gradients) algorithm[8] applies in pedestrian detection. Another approach to save the computational time in pedestrian detection is proposed[9] with the help of CBT (Cluster Boosted Tree) framework based on edge features. A two-stage classifier was used for fast pedestrian detection[10], features are prepared by analyzing HOG descriptors and are based on pixel orientation concept as well as multi-scaling levels. The results show a faster computation but still the occlusion detection and missing detection has a margin to improve. Bilgic B.[11] used rejection though Adaboost cascade framework approach for pedestrian identification. Feature selection is based on gradient direction histogram. They have attained an improved frame rate of 8 by using NVDIA

CUDA structure. A real-time pedestrian detection system is developed[12] and achieved correct recognition percentage of 62–100% at the cost of 0.3–5 false classifications per minute using multi-core vision approach.

Although with the rapid development of ConvNet (Convolution Neural Network), which provides top results for general object detection, research in the field of pedestrian detection still does not achieve a satisfactory result.[12–14] ConvNet propose an architecture that uses features from the last and second last layer for detection. A different line of work extends DPM (Deformable Part Model)[15] and mixtures of multiscale DPM[16]. Inspired by the success of R-CNN[17] for general object detection, a recent series of methods[18, 19] adopt a neural network for pedestrian detection. The Deep Parts method[18] applies the LDCF (Locally Decorrelated Channel Features) detector[20] to generate proposals and learns a set of complementary parts by neural networks. We observe that these proposers are stand-alone pedestrian detectors consisting of hand-crafted features and boosted classifiers.

2. Difficulties in Pedestrian Detection

In the past decade, through the joint efforts of scholars at home and abroad, pedestrian detection technology has made great breakthroughs in the algorithm, and has achieved fairly good results in a relatively fixed background such as an indoor environment. But in complex scenes, for example, the station, the square, the large shopping mall, and so on, or pedestrians are in a state of movement, stillness, change of attitude, and different degrees of mutual occlusion, all above bring difficulties to pedestrian detection and recognition. Next, the difficulties and problems of pedestrian detection are introduced in detail.

2.1 The problem of detecting in the complexity of the scene

In the traffic environment where there are pedestrians, a mixture between pedestrians and backgrounds is difficult to separate, mutual influence and occlusion between people, as well as the changes of illumination in the real scene, a large number of objects as hard negative examples who are similar to the pedestrian part contour, they all make it difficult for the system of pedestrian detector to accurately detect.

2.2 The problem of multi position change of pedestrians

Pedestrian targets are not rigid, while pedestrians may present a variety of different gestures, or walk or rest, or stand or squat. And there are also differences in the appearance of clothes between different pedestrians. How to design a robust detector for these changes is still a problem.

2.3 The problem of the real-time performance of the pedestrian detection system

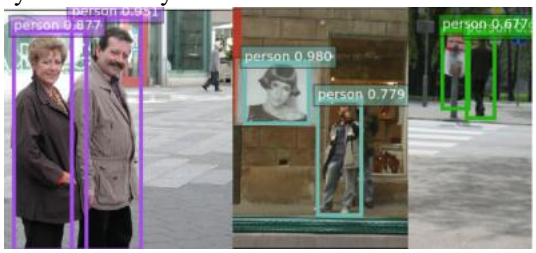
In practical application, the response speed of the detection system is often required. However, the actual detection and tracking system often need to deal with a large amount of data. Moreover, in order to satisfy the requirements of system robustness, algorithm building is often more complex, which has become a resistance to further enhance the real-time performance of the system.

2.4 Occlusion problem

In the real world, there is a lot of occlusion between pedestrians and objects in the detection environment. The existing image processing methods, the partial occlusion problem can be processed to a certain extent, but the effect is not very ideal, and it can't deal with the serious occlusion problem.

In conclusion, pedestrian detection in images is more challenging than detecting other general objects such as cars and faces because the appearance of people has a lot of fluctuations such as clothing, pose, or illumination. Because of this situation, there are many objects in the world that are difficult to separate from the pedestrians when the objects are small in pixels. In other words, pedestrian detector is not able to take advantage of scenario information for correct classification. Some results are shown in **Figure 1**. The detector is SSD trained on VOC datasets. Pedestrians and complex background bring lots of hard negative samples (usually appear in low pixels) such as trash can, traffic sign, pillar boxes and so on, all above have similar apparent features with pedestrians. The pedestrian detector without an extra semantic context is not able to

classify them directly.



(a) (b) (c)

Figure 1. (a) Pedestrian in small pixel is more likely to be missed. (b) Overlapped pedestrian is to be missed. (c) Objects similar to pedestrian appearance are easy for an error detection.

3. Algorithm SSD and Methods of Improving Pedestrian Detection Performance

3.1 Single-shot multiBox Detector SSD

Single-shot MultiBox Detector[21] is a deep learning network based on general object detector. It achieved a relatively high average precision of 74.3% on PASCAL VOC general object detection competition at high speed of 59FPS[21]. The authors propose SSD, a fully convolutional neural network which discretizes the possible output bounding boxes into a default set of bounding boxes at different scales and aspect ratios. The model predicts the object scores of each default bounding box, and regress the output bounding boxes' offsets to those default bounding boxes. SSD model uses VGG16[22] as a base network to generate feature maps.

The prior box of SSD was developed as a preview box for some targets, followed by the softmax classification and bounding box regression to get the real target location. To handle different object scale, original SSD imposes different aspect ratios and width for the default boxes. SSD uses additional feature layers at the end of the base network as showed above. Some of the generated feature maps are passed to a convolutional predictors to compute the confidence of each default bounding box in each of these feature maps and regress the bounding box offsets. During training, the default bounding boxes are matched to the ground truth bounding boxes.

3.2 The better hyper-parameters of the SSD for improving pedestrian detection proformance

To adapt different object scales, SSD imposes different aspect ratios for the default boxes, and denotes them as $\alpha_\gamma = \{1,2,3,1/2,1/3\}$. The SSD paper presents the width $\omega_k^\propto = s_k\sqrt{a^r}$ and height $h_k^\propto = s_k/\sqrt{a^r}$ for each default box. A default box whose scale is $s_k^{'} = \sqrt{s_k s_{k+1}}$ is also added, in 6 default boxes per feature map location. So there is an issue when using SSD for the task of pedestrian detection. If the SSD framework can't correctly cover most pedestrian boxes with the default box, many negative samples will be generated which would lead to a bad training result. So the default boxes scales and aspect ratios represent the hyper-parameters of the SSD model that should be chosen wisely based on the dataset objects sizes and aspect ratios. An analysis is showed as below based on Caltech pedestrian dataset[23]. This dataset is split into 6 sets for training and 5 sets for testing. Here we selected 16334 labeled pedestrians in the Caltech pedestrian training set for statistical analysis.

The ratio α_γ (width to height) is just an important parameter for pedestrian rectangles. **Figure 2** shows the histogram of the ratio. The mean and the variance of the ratio are 0.398, 0.013, respectively. As can be observed from the graph, the aspect ratio of most pedestrian samples in the training data values fluctuates within a small range of 0.398, and the ratio subject to normal distribution.

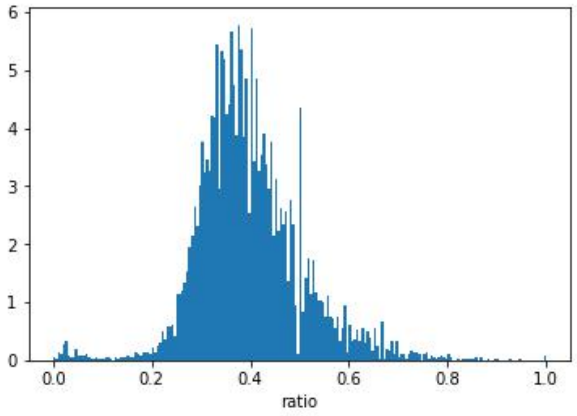


Figure 2. The histogram of the ratio.

The scale of width was statistically analyzed too.

Figure 3 displays the histogram of the width. The mean and the variance of the ratio are 26.625 and 562.418 respectively. The large variance predicts great values fluctuation in pedestrian scale, which also proves the necessity to set a different scale in SSD.

Based on the analysis of the data, we impose the same aspect ratios for the default boxes, and set them as $\alpha_\gamma = \{0.39\}$. Compared with the original parameters in SSD , the scale of the default box decreased by 10% so as to fit the analysis scale.

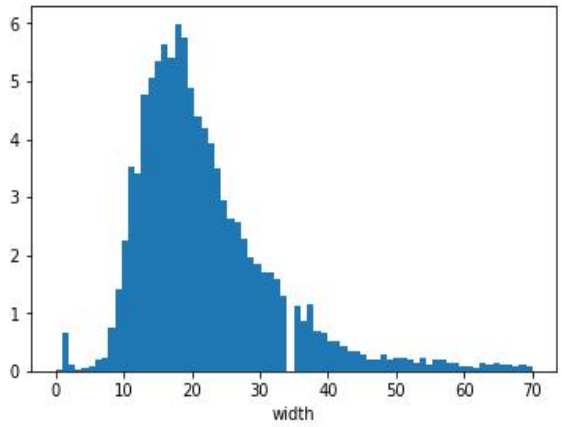


Figure 3. The histogram of the width scale.

We made a group of comparative experiments. The Caltech data set is used for the training process. To verify the generalization ability of the algorithm, SSD is tested with the USC dataset. In the training and test procedure, the size of each image sample is normalized to 300*300 pixels. For training of the SSD framework, we selected the data set from the total Caltech training set by a manner of jump, and then 3750 pictures were selected as the training set. In the test procedure, we chose 99 images sample from USC data set, images are chosen with pedestrians in various orientation, pose, color, and distance from the camera to increase robustness. The results of the pedestrian detection are shown using bounding box drawn around detected pedestrian in the given image.

During the training phase, we trained the SSD model in the Lenovo computer, equipped with the NVIDIA GTX1050 graphics card, and installed CUDA8.0, CuDNN, each batch size is 6 samples, iterates 120,000 times. The initial learning rate is 1e-4, which is reduced by 10% every 7000 iterations. And the stochastic gradient descent method is used for optimization. The training time is about 25 hours. Accuracy Performance of SSD Model is shown in **Table 1**.

Table 1. SSD model performance over the default configration

	AP	Improvement
SSD (training from scratch)	59.72%	–
+ finetuning	63.46%	3.74%
+ hyper-parameters	67.39%	3.93%

3.3 The better backbone network of the SSD for improving pedestrian detection

There is one backbone network in SSD framework, which is utilized to extract the features from the images. We would like to know whether the different backbones affect the pedestrian detection performance, especially on hard negatives examples without reducing accuracy. Then, we designed an experiment to verify the impact of infrastructure on pedestrian detection. One took the VGG16 and the other the Resnet50. The two networks are introduced as follows.

VGGNet[22] is proposed by Visual Geometry Group of the University of Oxford, which is the first and second task of the location task in ILSVRC-2014. Its outstanding contribution is to demonstrate the use of very small convolution kernel (3*3). Increasing network depth can effectively increase the efficiency of the model, and VGGNet has good generalization ability to other datasets. Convolution neural network has become a common tool in the field of computer vision nowadays. So many people try to improve the AlexNet proposed in 2012 to achieve better results. For example, the best performing ZFNet in -2013 in ILSVRC uses smaller convolution (receptive window size) and smaller step length (stride) in the earliest layer of layers. Another strategy is to multiscale the intensive training and testing of the entire image. VGGNet emphasizes another important aspect of the design of convolution neural networks - depth.

ResNet[24] is the best paper of CVPR in 2016 which was proposed to solve the problem that how to make it the convergence of the very deep network. A framework for residual learning is proposed in this paper. Then

compared with VGG briefly, the 152 level residual network is 8 times deeper than VGG, but it is lower than the VGG complexity. Of course, the performance on ImageNet is better than that of VGG, and it is the champion of the 2015 ILSVRC classification task.

To conduct a relatively fair experiment, the VGG model and ResNet50 model are both been pre-trained on the ImageNet which is the largest database of image recognition in the world. And then we train SSD on the dataset of PASCAL VOC dataset 07 12 which contains twenty pieces of objects including pedestrian. To verify the generalization performance of the algorithm, We chose 118 pictures as the test sets, which mostly come from the INRIA Person dataset.

The result is shown in Table 2. The first line is the thresh of confidence and the rest lines stand for the detection precision rate in the test sets.

Through **Table 2** and **Figure 4**, we can conclude that the two basic networks are almost the same in AP. Therefore, changing the backbone network will not bring about great fluctuation of AP. The total time of VGG16 and ResNet50 is 12.611 seconds, 10.856 seconds respectively, and the average time of each image is 0.107 seconds, 0.092 seconds.

Table 2. The performance of VGG16 and ResNet50

Thresh	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
AP（VVG 16）	99.15	99.15	98.31	97.46	94.92	93.22	91.53	88.98	83.05
AP (ResNet 50)	99.15	98.31	98.31	98.31	98.31	94.07	88.14	85.60	72.88

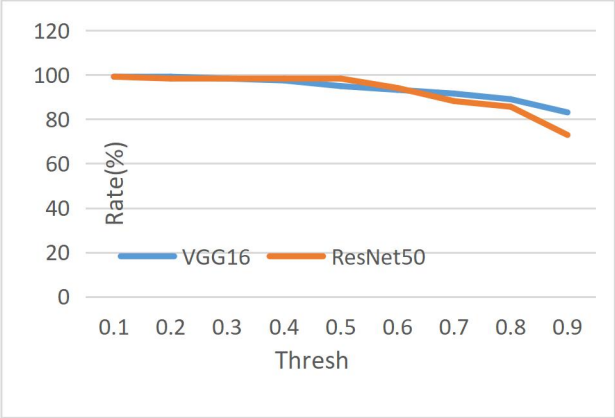


Figure 4. Backbone Network performance.

3.4 Loss function design for SSD

Since SSD produces 8832 candidate rectangles, there are few pedestrians on each image, which leads to an extremely unbalanced ratio of positive and negative samples. Meanwhile, most of the negative samples are easy example.

There is a large degree of positive and negative samples imbalance in the training process on the one-stage detector. SSD take a strategy of OHEM (Online Hard Example Mining)[25] to alleviate this problem. Focal Loss[26] was proposed to solve the imbalance of positive and negative samples on the one-stage detector. so the focal loss is added to the experiment for an exploration.

We modified the default cross entropy loss function with focal loss , which was proposed to solve the positive and negative samples imbalance of general object detection to compare the performance of OHEM and Focal Loss. The experiment was trained on Caltech pedestrian dataset and tested on USC dataset.

From the **Table 3**, we could see it could improve our AP about 2 points, which could prove effectiveness of the loss function.

Table 3. The contrast about loss function

Method	**AP**	**Improvement**
OHEM	63.46%	–
Focal-loss	65.83%	2.37%

4. PS-GAN Design for Training Sets Generation

GAN (Generative Adversarial Net) has been successfully applied in image synthesis field (such as DCGAN, DSGAN, VS-GAN, etc.)[28–31]. Its basic idea is to input a noise image, generate a fake image with Generator to deceive the Discriminator, and Discriminator tries to distinguish the false image from the real image. In the course of training, the generator will become stronger and stronger, and the generated pseudo images are almost the same as real images. Our goal is to make the fake pedestrian image generated by GAN assist the training process of pedestrian detector, so as to improve the performance of pedestrian detector.

4.1 The structure of PS-GAN

A basic idea of this method is to train a GAN so that it can synthesize scene-blended images. The whole structure of PS-GAN is shown as **Figure 5**.

Generator (G): First, we replace a pedestrian box in a real image of a scene graph with noise (the purpose is to generate a pedestrian in this box), and then send it to generator G. The output is a generated scene graph image.

Discriminator (D): It consists of two parts: The discriminator is applied to classify between real and synthesized pair to learn the background context in the noise box. The discriminator learns to classify the real and synthesized pedestrian with the noise box.

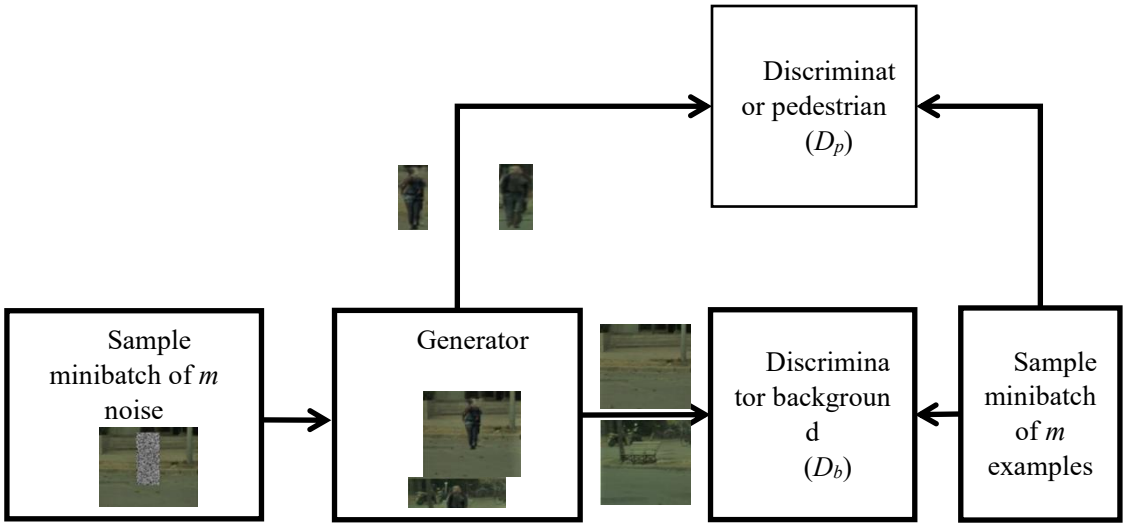


Figure 5. The structure of PS-GAN.

4.2 The loss function of PS-GAN

The model of PS-GAN consists of two adversarial training procedures $G \leftrightarrow D_b$ and $G \leftrightarrow D_p$. The adversarial training between G and D_b can be formulated as:

$$\mathcal{L}(G, D_b) = E_{y \sim p_{gt.image(y)}}[(D_b(y) - 1)^2] + E_{x,z \sim p_{noise.image(x,z)}}[(D_b(D(x,z)))^2] \quad (1)$$

Where x is the image with the noise box and y is the ground truth image. We use a least square loss to take place of the original GAN loss[28,29].

To let G to generate realistic pedestrians within the box z in the input image x, another adversarial training procedure is conducted between G and D_p:

$$\mathcal{L}\left(G, D_p\right) = E_{y_p \sim p_{pedestrian(y_p)}}\left[log\, D_p(y_p)\right] + E_{z \sim p_{noise(z)}}\left[log\left(1 - D_p(G(z))\right)\right] \tag{2}$$

Where z is the noise box in x and y_p is the cropped pedestrian in the ground truth image y.

The training procedure of GAN can be stable when the traditional loss is used. In this paper, we apply ℓ_1 loss to balance the differences between the generated image and ground image y:

$$\mathcal{L}_{\ell_1}(G) = E_{x,z \sim p_{noise.image}(x,z), y \sim p_{gt.image(y)}}[\, \| y - G(x,z) \|_1] \tag{3}$$

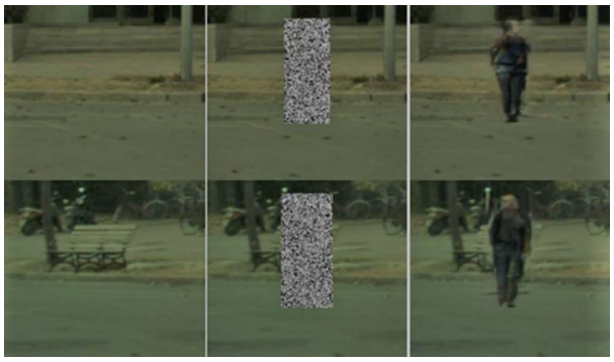
The final loss function is defined by combined the losses previously defined. Which λ controls the relative importance of ℓ_1 loss.

$$\mathcal{L}\left(G, D_b, D_p\right) = \mathcal{L}\left(G, D_b\right) + \mathcal{L}\left(G, D_p\right) + \lambda \mathcal{L}_{\ell_1}(G) \tag{4}$$

4.3 Cityscapes

The cityscapes dataset is a new large-scale dataset that contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, with high quality pixel-level annotations of 5 000 frames in addition to a larger set of 20 000 weakly annotated frames. Cityscapes has relatively high resolution pictures and contains more pedestrians with rich variety, which is suitable to train our GAN model.

We cropped 512 x 512 patches around the labeled pedestrians from the original 1024 x 2048 images. There are some labeled pedestrians which are too small or partially blocked by pedestrian or other objects. So we reserved all the cropped images whose bounding box with the width range from 29 to 64 and height range from 80 to 130. After that we obtain 1558 images and randomly select 100 images of them as the test dataset. Then those noise images are taken as the training data for PS-GAN. **Figure 6** shows some generated samples by the generator.



(a)original (b) noise (c) generate

Figure 6. Systhesis pedestrian in backgroud images.

4.4 Experiment analysis

In this section, we combined the real data and the data generated by PS-GAN to train SSD detector to analyze the effects of data augmentation. To demonstrate how the augmented synthetic images can help boost the performance of the SSD model, we train three SSD detectors (VGG-16 based model). The baseline detector is trained on the PS-GAN original 1558 training images, and the other detectors are trained on those images adding synthetic pedestrians from PS-GAN. All the detectors are tested on the 500 testing images and the average precision (AP) are from the best performance when all the models converge. **Table 4** shows the performance of the PS-GAN.

Table 4. The result of PS-GAN, using different settings to train the SSD

Data	PS-GAN
1558 real images	59.76 %
+ 1000 synthetic pedestrians	60.32 %
+ 2000 synthetic pedestrians	61.18%

As shown in **Table 4**, all detectors were pretrained on ImageNet datasets. The baseline detector using 1558 real images could achieve 59.76% of the average precision(AP) for pedestrian detection. By adding the synthetic pedestrians, the AP rate can be improved.

5. A Specific Application of Motor Vehicle Assisted Driving

In the course of vehicle driving, an excellent driver can accurately determine which pedestrians on the road will threaten the vehicle according to the current velocity of the vehicle, thereby improving the safety of driving. In view of the task of vehicle auxiliary driving system, we propose a simple and effective method to locate the target of interest pedestrians, which will improve the pertinence of the pedestrian detection targets.

To locate the target of interest pedestrians who will threaten the vehicle mostly, we have assigned a speed factor to the SSD detector so that the detector can reasonably detect the necessary pedestrians based on the velocity of the vehicle and exclude unnecessary pedestrians. The mathematical description of the procedure is as follows:

The v_k stands for the factor of velocity that has a relevance with $d_{i\sim k}$, where $d_{i\sim k}$ is the distance range under current speed factor, (5) presents a mathematical relationship, where we can calculate this distance range of interest based on the speed factor. **Table 5** shows one situation we collected about the relationship between v_k and $d_{i\sim k}$.

$$v_k \rightarrow d_{i\sim k} \;\; ; \;\; \{v_k \geq 0\} \tag{5}$$

First we need to calculate the focal length of the camera. The formula is shown in (6), where h_k is the height of the pedestrian in image and H_k stands for the pedestrian's true height. d_i is the distance range under current speed factor.

$$\lambda = (h_k * d_i) \,/\, H_k \;; \{v_k \rightarrow d_{i\sim k}\} \tag{6}$$

Because pedestrians on the road have different height, while the algorithm need to fix the height of each pedestrian, so we take a maximum and minimum height of the pedestrian and then roughly estimate a range of distances for pedestrians in the image. The pedestrian detector will give each pedestrian's bounding boxes $\{x_k, y_k, w_k, h_k\}$, $k \in R$ to locate the target of interest, we select the height of the bounding boxes h_k according to the relationship between the speed v_k and the distance $d_{i\sim k}$ of interest, formula (7) shows the relationship of h_i and d_i. So we could make the fact v_k, $h_k \rightarrow \Omega$, where Ω is the range containing a set of targets that the detector is interested.

$$h_i = (\lambda * H_k) \,/\, d_i \tag{7}$$

5.1 The experiment for the target of interest in motor vehicle assisted driving

We have collected a set of data of some outstanding drivers on the road who can output the area of interest through the velocity of the vehicle.

Table 5. The experiment parameter of velocity and distance

Velocity(km/h)	Distance (m)
0~20	0~7.32
20~40	7.32~20.92

The height of pedestrian is different. Schilling[27] made a statistical analysis of human height, which pointed out that male and female height enjoy their own normal distribution. **Figure 7** shows the normal distribution graph for male and female. We used the Caltech data where we chose 47 images set for the invalidation. The speed factor was imbedded in the images with the value of 0~20. Heights of pedestrian are arranged from 165cm to 175cm.

5.2 The result analysis of the experiment for the target of interest

The size of Caltech data images is 640*480 pixel, through (6) we can calculate the $\lambda \cong 536.11$, this parameter should be calculated for other tasks. According to λ, we can figure out the relationship between the

velocity and the height h_k of the bounding box. It was the data calculated by our experiment. The contrastive result is shown in **Table 6**, which presents a nice performance improvement.

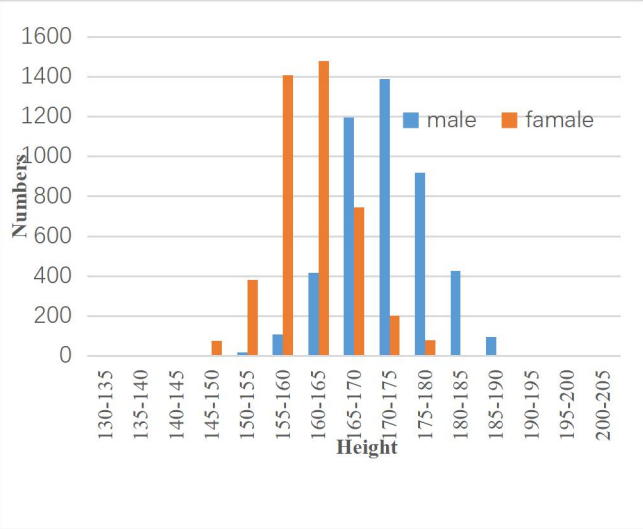


Figure 7. The height distribution of male and female.

Table 6. The contrast result for Target of interest

Method	Velocity(km/h)	
	0-20	20-40
Without TARGET OF INTEREST	0.584	0.674
With TARGET OF INTEREST	0.707	0.740

As is shown in **Figure 8**, we could see the vehicle is at a relatively stationary speed, all the pedestrians in the left column images are detected with red rectangles, while some pedestrians do not need to be detected. The right column exhibits a fact that the pedestrian detector excludes all pedestrians within the range of 7.32 meters who not need to be detected, the target of interest is in the yellow rectangle.

(a) (b)

Figure 8. (a) Column shows normal detection before appling Target of interst. (b) Shows the appling for Target of interst.

From **Table 6**, we could see a great improvement in AP under the fixed velocity. It performed very admirable in our experiment. The experimental results showed that our method can effectively exclude unnecessary detection target, this change direction excluded some missed detection and improved the detection accuracy.

6. Conclusions

We have presented extensive and systematic experimental evidence on the pedestrian detection performance based on SSD and PS-GAN. The experimental results demonstrate that our methods as simple common tricks can improve pedestrian detection performance in varying degrees.

We have shown that with pre-training and a best hyper-parameters for pedestrian can reach significant performance on this task. Interestingly we compare and analyze the performance of Focal Loss and OHEM algorithms on the single-stage detector SSD, and this result is quite insensitive to the model parameters (the two models own different parameters and architectures).

Our experience with different approaches that aim to improve pedestrian performance shows good promise on pedestrian detection, and reported best practices do transfer to said task. The proposed PS-GAN method takes into account the pedestrian characteristics and adds training samples to improve the accuracy of pedestrian detection.

At last, we use the pedestrian detector to simulate a specific application of motor vehicle assisted driving which would make the detector focus on specific pedestrians according to the velocity of the vehicle. The application focus on the relationship of the velocity of the vehicle and the distance between pedestrian and vehicle, it can filter out pedestrians whom are not interested, the accuracy will be significantly improved by applying it. But it also has some inner defects, such as its effectiveness depends heavily on the accuracy of the detection frame, pedestrians do not have a uniform height and so on. Future work could focus on the improvement effective regional propose and the structure of the system itself, small targets for pedestrians are also a necessary problem to be solved, so as to increase the overall performance, with the aim of combining state-of-the-art accuracy and real-time processing.

Acknowledgment

The work is partly supported by Beijing Educational Science Planning (Grant No. CADA18069) in 2018.

References

1. Dollar P, Wojek C, Schiele B, *et al*. Pedestrian detection: a benchmark. Proc. conf. on Computer Vision & Pattern Recognition, 304-311, 2009.
2. Piotr Dollár, Tu Z, Perona P, *et al*. Integral channel features. British Machine Vision Conference. DBLP 2009.
3. Urtasun R, Lenz P, Geiger A. Are we ready for autonomous driving? The KITTI vision benchmark suite. IEEE Conference on Computer Vision & Pattern Recognition 2012.
4. Benenson R, Omran M, Hosang J, *et al*. Ten years of pedestrian detection, what have we learned? 2014.
5. Zhang L, Lin L, Liang X, *et al*. Is faster r-cnn doing well for pedestrian detection? 2016.
6. Kakadiaris IA, Metaxas D. 3D human body model acquisition from multiple views. International Conference on Computer Vision. IEEE 1995.
7. Rohr K. Towards model-based recognition of human movements in image sequences. CVGIP: Image Understanding 1994; 59(1): 94-115.
8. Dalal N, Triggs B. Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision & Pattern Recognition 2005.
9. Sermanet P, Kavukcuoglu K, Chintala S, *et al*. Pedestrian detection with unsupervised multi-stage feature learning. Computer Vision & Pattern Recognition 2013.
10. Ye Q, Jiao J, Zhang B. Fast pedestrian detection with multi-scale orientation features and two-stage classifiers. IEEE International Conference on Image Processing. IEEE 2010.
11. Bilgic B, Horn BKP, Masaki I. Fast human detection with cascaded ensembles on the GPU. Intelligent Vehicles Symposium 2010.
12. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. International Conference on Neural Information Processing Systems. Curran Associates Inc 2012.
13. Russakovsky O, Deng J, Su H, *et al*. Imagenet large scale visual recognition challenge. International Journal of Computer Vision 2014; 115(3).
14. Sermanet P, Kavukcuoglu K, Chintala S, *et al*. Pedestrian detection with unsupervised multi-stage feature learning. In CVPR 2013; 1, 2, 5.
15. Felzenszwalb P, Mcallester D, Ramanan D. A discriminatively trained, multiscale, deformable part model. Cvpr 2008; 8: 1-8.
16. Felzenszwalb PF, Girshick RB, Mcallester D, *et al*. Object detection with discriminatively trained part-based models. IEEE Transactions on Software Engineering 2010; 32(9): 1627-1645.
17. Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society.
18. Tian Y, Ping L, Wang X, *et al*. Deep learning strong parts for pedestrian detection. IEEE International Conference on Computer Vision 2016.
19. Tian Y, Ping L, Wang X, *et al*. Pedestrian detection aided by deep learning semantic tasks 2015.
20. Nam W, Dollar P, Han JH. Local decorrelation for improved pedestrian detection. NIPS 2014.
21. Liu W, Anguelov D, Erhan D, *et al*. SSD: Single shot multibox detector 2015.
22. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Computer Science 2014.
23. Wojek C, Dollar P, Schiele B, *et al*. Pedestrian detection: an evaluation of the state of the art. IEEE Transactions on Pattern Analysis & Machine Intelligence 2012.
24. He K, Zhang X, Ren S, *et al*. Deep residual learning for image recognition 2015.
25. Shrivastava A, Gupta A, Girshick R. Training region-based object detectors with online hard

example mining 2016.
26. Lin TY, Goyal P, Girshick R, *et al*. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis & Machine Intelligence 2017; PP(99), 2999-3007.
27. Schilling MF, Watkins AE, Watkins W. Is human height bimodal? American Statistician 2002; 56(3): 223-229.
28. Goodfellow Ian J., Pouget-Abadie Jean, Mirza Mehdi, *et al*. Generative Adversarial Networks 2014. eprint arXiv:1406.2661.
29. Qin Pengda, Xu Weiran, Wang William Yang. DSGAN: Generative adversarial training for distant supervision relation extraction. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics 2018; 496-505.
30. Zheng K, Wei M, Sun G, *et al*. Using vehicle synthesis generative adversarial networks to improve vehicle detection in remote sensing images. ISPRS Int. J. Geo-Inf. 2019; 8: 390.
31. Wei M, Zheng K, Li S, *et al*. The target detection based on YOLOv3 and PVSGAN. Basic& Clinical Pharmacology&Toxicology 2019; 074(125): 45-45.