Peer-reviewed version available at *Appl. Sci.* **2020**, *10*, 1866; doi:10.3390/app10051866

Article

Pedestrian Detection Based on Two-Stream UDN

Wentong Wang [1,2,3], Lichun Wang [1,2,3], Xufei Ge[4] and Jinghua Li [1,2,3,] *, Baocai Yin [1,2,3]

[1] Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology, Beijing. 100124, China
[2] Beijing Artificial Intelligence Institute, Beijing University of Technology, Beijing. 100124, China
[3] Faculty of Information Technology, Beijing University of Technology, Beijing. 100124, China
[4] CITIC Guoan Broadcom Network Co., Ltd. Beijing, 100176, China
* Correspondence: lijinghua@bjut.edu.cn

Featured Application: This paper can be applied to autonomous vehicle and driving assistance system.

Abstract: Pedestrian detection is the core of driver assistance system, which collects the road conditions through the radars or cameras on the vehicle, judges whether there is a pedestrian in front of the vehicle, supports decisions such as raising the alarm, automatically slowing down or emergency stopping to keep pedestrians safe, and improves the security when the vehicle is moving. Suffered from weather, lighting, clothing, large pose variations and occlusion, the current pedestrian detection still has a certain distance from the practical applications. In recent years, deep networks have shown excellent performance for image detection, recognition and classification. Some researchers employed deep network for pedestrian detection and achieve great progress, but deep networks need huge computational resources which make it difficult to put into practical applications. In real scenarios of autonomous vehicle, the computation ability is limited. Thus, the shallow networks such as UDN (Unified Deep Networks) is a better choice since it performs well on consuming less computation resources. Base on UDN, this paper proposes a new deep network model named as two-stream UDN, which augments another branch for solving traditional UDN's indistinction of the difference between trees / telegraph poles and pedestrians. The new branch accepts the upper third part of the pedestrian image as input, and the partial image has less deformation, stable features and more distinguished characters from other objects. For the proposed two-stream UDN, multi-input features including HOG feature, Sobel feature, color feature and foreground regions extracted by GrabCut segmentation algorithms are fed. Compared with the original input of UDN, the multi-input features are more conducive for pedestrian detection since the fused HOG features and significant objects are more significant for pedestrian detection. Two-stream UDN is trained through two steps: First, the two sub-networks are trained until converge; then we fuse results of the two subnets as the final result and feed it back to the two subnets to fine tune network parameters synchronously. To improve the performance, Softplus is adopted as activation function to obtain faster training speed, and positive samples are mirrored and rotated with small angle to make positive and negative samples more balanced.

Keywords: Pedestrian detection; Unified Deep Net; Two-stream nets; Network training

1. Introduction

Pedestrian detection is an important research field in computer vision. In recent years, it has become more and more widely used in vehicle-assisted driving, intelligent video surveillance, and human behavior analysis. According to the Global Status Report on Road Safety 2018 published by the World Health Organization [1], 1.3 million people die in a year in traffic accidents and

Peer-reviewed version available at Appl. Sci. 2020, 10, 1866; doi:10.3390/app10051866

distributions of traffic deaths of different types of road users in WHO regions are shown in Figure 1. Worldwide, 23% of the road traffic fatalities are pedestrians.

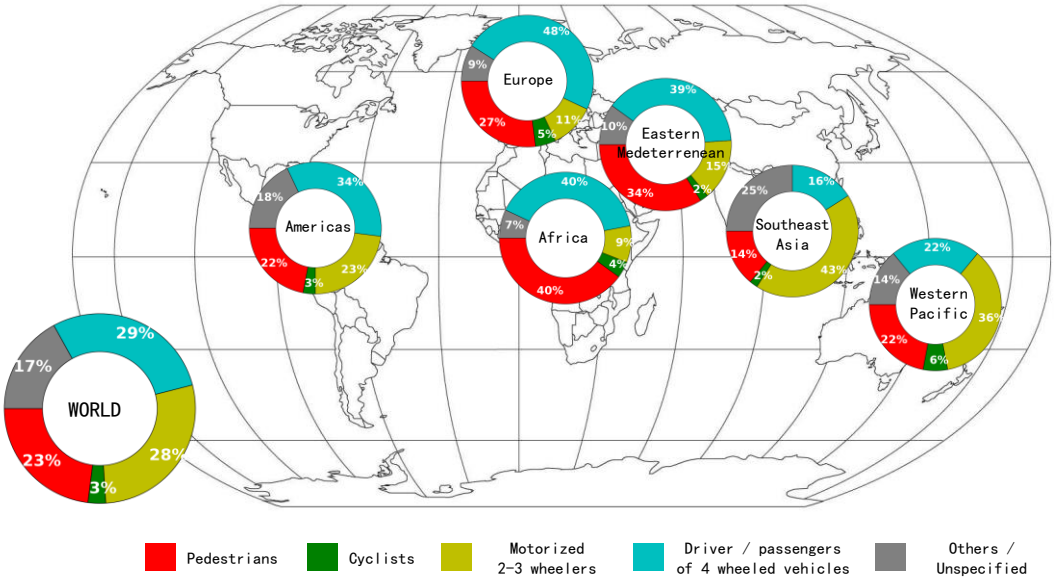


Figure 1. Distribution of deaths of road user in 2018 reported by the WHO

In order to reduce occurrence of traffic accidents and protect safety of pedestrians, major research institutions have conducted research on intelligent vehicle assisted driving systems. The system includes a pedestrian detection system, which uses on-board camera to obtain road condition information, and applies intelligent detection algorithm to detect pedestrians in the driving environment, reminds the driver to keep a safe speed, and promptly reports whether a collision may occur or performs deceleration and automatic braking.

As a hot issue in the field of computer vision, pedestrian detection has attracted the attention of scholars in the mid-1990s. Difficulties in pedestrian detection includes the following: complex background in reality, the influence of weather or light, diversity of pedestrian poses, diversity of pedestrian clothing, occlusion of pedestrian and other objects, and different camera viewpoints. In order to solve the above problems, researchers have proposed a large number of algorithms to promote the effectiveness of pedestrian detection.

2. Related works

The key to pedestrian detection is extracting pedestrian features, but pedestrian is non-rigid object complicated to represent. According to different ways of feature extraction, pedestrian detection methods can be divided into three types, template matching, statistical learning, and deep nets.

2.1. Template matching based pedestrian detection

The pedestrian detection algorithm based on template matching aimed to establish a pedestrian target template database for different types of pedestrians. The template can be a pedestrian outline or a grayscale image. When detecting an image, the template features of the input image are calculated, and then search in a pedestrian template database to find whether a corresponding pedestrian template matches the input template feature successfully.

The contour-based hierarchical matching algorithm proposed by Gavrila [2,3] is a typical template matching algorithm. The algorithm firstly used layered matching method to lock candidate area by using contour features of human body, compute distance transformation image of the candidate area and Chamfer distance between distance transformation images and pedestrian

templates. The second step used radial basis functions (RBF) with Chamfer distance to verify whether the candidate area is pedestrian. The candidate area includes person, so the algorithm belongs to overall template matching method.

In order to tackle occlusion and the variation of human pose, Shashua [4] proposes an algorithm based on human part template matching. The algorithm divided human body into nine parts with overlapping regions, and built nine types of pedestrian templates. Wu and Nevatia [5] divided the human body into three parts, head with shoulders, torso and legs, and used Edgelet features for part detection. Each Edgelet is composed of a set of edge points to describe outline of part of the human body.

Pedestrian detection based on template matching algorithm is relatively simple to calculate, but the algorithm requires a pre-designed pedestrian template database. The human body is a non-rigid object, and the pose changes are more complicated. Therefore, the algorithm has certain limitations. At the same time, the template matching algorithms only consider outline information of human body and ignores details such as the skin color and clothing of the human body. When the contour feature of pedestrians is relatively fuzzy, the performance would drop.

2.2. *Statistical learning based pedestrian detection*

Methods based on statistical learning refers to learning a classifier through a series of training data and classified input regions with the learned classifier. Feature extraction and classifier design are core techniques of pedestrian detection based on statistical learning.

In feature extraction stage, the key issue is extracting discriminative pedestrian features. A good feature should not only capture information different from other classes, but also maintain the stability of differences within class. Current feature acquisition methods for pedestrian detection include hand-crafted features and learned features. Hand-crafted features commonly include Haar-like features [6], SIFT features [7], HOG features [8], and variations or combinations of these features. Using HOG is an important milestone in pedestrian detection, which highly promotes pedestrian detection effect. CSS (color self-similarity) characterizing interrelationships between local block features [9] is combined with HOG to improve detection performance greatly. Integral Channel Features [10] also achieved good detection results in pedestrian detection. Felzenszwalb et al. DPM (Deformable Part Model) [11,12], as a spring deformation model [13], is effective for solving pose changing in pedestrian detection.

In classification stage, commonly used classifiers include SVM[17], Random Forest, Probabilistic Model, Neural Network[14-16]. Combination of HOG and SVM is a classic algorithm in the history of pedestrian detection.

Benenson et al. [18] suggested that classifier has less impact on pedestrian detection, and extracted features are more important. Pedestrian detection based on statistical learning adapts to images with simple backgrounds and less occlusion, but its effects need to be improved for images with complex backgrounds and large pose changing [14], so more robust features must be found.

2.3. *Deep learning based pedestrian detection*

As an effective feature learning method, deep learning has made breakthrough in applications such as computer vision, data mining, and speech recognition. In recent years, researchers have conducted in-depth research on pedestrian detection based on deep learning, and have achieved abundant results. DBN-ISOL[14] was proposed for part detection, which performs well on pedestrian detection in the presence of occlusion. ConvNet[19] consists of three convolutional layers, features obtained by the second and third layer are fused as input to fully connected layer to finish pedestrian detection. DBN-Mut[20] extended mutual visibility based on DBN-ISOL, which focuses on the situation where a pedestrian is partially blocked by another pedestrian during pedestrian detection. UDN[21] was constructed based on CNN, Part detection, Deformation model, and Visibility reasoning. It makes full use of the advantages of DBN-ISOL and DBN-Mut, and combines CNN and BP deep networks for pedestrian detection. SDN[22] combined feature learning, saliency mapping, and mixed feature representation in a cascade structure, in which a switchable RBM layer is

introduced on the traditional CNN to selectively combine different features. The above methods usually include two or three layers in the deep net.

To learn more sophisticate features, researchers implemented deeper networks to extract features, which usually include at least 16 layers. UDN was extend to a very deep net by using VGG16 and fast RCNN to obtain features of different body part for pedestrian detection [23]. RPN and faster RCNN were used simultaneously to detect and segment pedestrians in image [24], the backbone network is VGG16. PCN [25] uses a LSTM for part semantic learning, a RPN for region proposals, and a Maxout of different adaptive scale selection and the backbone network is also VGG16. By employing an attention network to the baseline faster RCNN, better detection was obtained [26]. Zhang et al. [27] and Song et al. [28] employed ResNet with 50 layers for pedestrian feature extraction, and obtained state-of-the-art results.

However, in real scenarios of auto-driving, the hardware computing ability is limited. Networks with too much layers are not practical, so shallow network such as UDN is worth consideration.

3. Two-stream UDN for Pedestrian Detection

For pedestrian detection, feature extraction, part deformation handling, occlusion handling and classification are four important components. However, existing methods learn or design these components individually or sequentially. In order to maximize their strengths through cooperation, UDN[21] formulated the four components into a joint deep learning framework which is a shallow network having two convolution layers.

3.1. Improved UDN

The UDN process images in YUV space. The three channels fed to UDN include two images and one feature image, shown in the upper row of Figure 2. The first channel is Y channel of input image. The second channel consists of 4 small images and each is one fourth of the original image, three of them are Y, U and V channel of the original image, the fourth part is blank image. The third channel also consists of 4 small images and each is one fourth of the original image, three of them are Sobel edge features computed with Y, U and V channel of the original image, the fourth part is achieved by maximizing the three edge images on pixel-wise. It can be seen that, for some images the UV channels may be too plain to provide enough information for recognition, and corresponding edge maps are nearly zeros, which is not benefit for feature extractions.

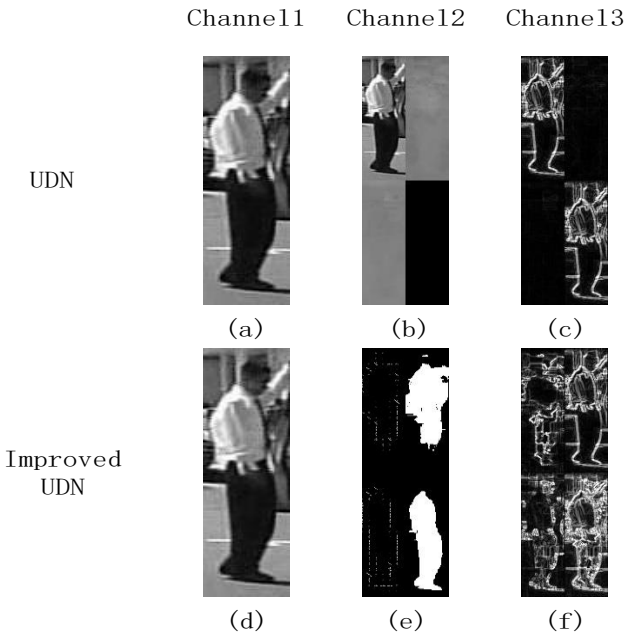


Figure 2. Input channels for UDN and improved UDN. (a) Y channel in YUV color space. (b) Concatenation of Y channel, U channel and V channel with zero padding. (c) Concatenation of four

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2020 doi:10.20944/preprints202001.0029.v1

Peer-reviewed version available at Appl. Sci. 2020, 10, 1866; doi:10.3390/app10051866

Sobel edge maps. Former three are obtained from three channel image in YUV color space, the fourth edge map is obtained by choosing the maximum magnitudes from the first three edge maps. (d) V channel in HSV color space. (e) Concatenation of two HOG feature maps and two Grabcut feature maps. (f) Concatenation of four Sobel edge maps. Former three are obtained from three channel image in HSV color space, the fourth edge map is obtained by choosing the maximum magnitudes from the first three edge maps.

In this paper, we employ HSV color space for feature extraction and provide more shape information for network. Lower row of Figure 2 shows new input for networks. The three channels are as follows: (1) The first channel is the V channel extracted from the original image after HSV colour space conversion; (2) The second channel is divided into four blocks, which are HOG features of HSV image, GrabCut regions of HSV image, HOG features of RGB image, and GrabCut regions of RGB image. (3) The third channel consists four blocks. The first three blocks are HSV edges, which are calculated by the 5×5 Sobel operator for the three channels of the HSV image respectively. The fourth block is achieved by maximizing the three above edge images on pixel-wise. Compared with UDN, our proposed feature combination provides more shape information.

Figure 3 shows improved UDN, which has same structure with UDN but adopts different input features and different activation function. Details are as following:

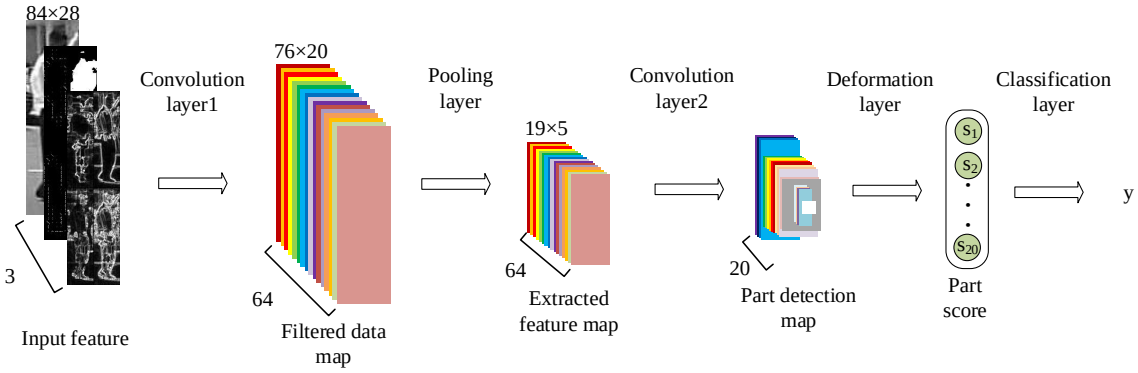


Figure 3. Improved UDN

1) Convolution layer 1: The input of the first convolution layer has three components $X_i \in R^{84\times28}, i = 1,2,3$, which are corresponding to channels in Figure 2. There are 64 convolution kernels $\{W_j | W_j \in R^{9\times9\times3}, \mathrm{j} = 1, \ldots, 64\}$ are utilized for extracting features. With each convolution kernel W_j, a filtered data map $A_j \in R^{76\times20}$ is computed as follows:

$$A_j = Softplus(b_j + \sum_{i=1}^{3}[W_j]_{:,:,i} * X_i) \quad (1)$$

where $[W_j]_{:,:,i}$ represents the i^{th} slice of three-order tensor W_j, * represents convolution operator, b_j represents bias parameters obtained by random initialization. Before training, W_j is initialized with Gabor filter.

2) Pooling layer: Calculate the average value of pixels in each 4 × 4 neighborhood on each filtered data map and obtain extracted feature maps $\{B_k | B_k \in R^{19\times5}, \mathrm{k} = 1, \ldots, 64\}$, which is computed as follows:

$$B_k = AveragePool(A_k, 4, 4) \quad (2)$$

where k represents the number of the extracted feature maps.

3) Convolution layer 2: The second convolution layer has 20 part-based filters $F_n \in R^{p_n\times q_n\times64}, n = 1, \ldots, 20$ with different sizes. The filters are same with UDN. The part detection map C_n is computed as follows:

$$C_n = Softplus(u_n + \sum_{m=1}^{64}[F_n]_{:,:,m} * B_m) \quad (3)$$

where n represents the number of the filtered data maps, $[F_n]_{:,:,m}$ represents the m^{th} slice of three-order tensor F_n, * represents convolution operator, u_n represents bias parameters obtained by random initialization. Before training, F_n uses Gabor filter for initialization.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2020 doi:10.20944/preprints202001.0029.v1

Peer-reviewed version available at Appl. Sci. 2020, 10, 1866; doi:10.3390/app10051866

4) Deformation layer: The deformation layer is same with the UDN [21] and returns score s_P for the p^{th} part, $\{s_P|p = 1, \dots, 20\}$.
5) Classification layer: The classification layer is same with the UDN [21] and estimates label y of input image, pedestrian or non-pedestrian.

The improved UDN adopts Softplus as activation function, different from Sigmoid used in UDN [21]. Otherwise, ReLU is a commonly used unsaturated activation function in deep learning in recent years since it can solve vanishing gradient problem, but its derivative is discontinuous. Figure 4 shows comparison of Sigmoid, ReLU and Softplus, we can see that Softplus is a non-linear continuously differentiable function approximating to ReLU smoothly.

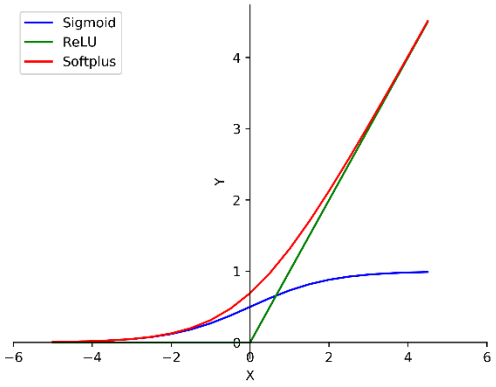


Figure 4. Comparison of Sigmoid, ReLU and Softplus

Softplus has similar characteristics to the activation frequency function of brain neurons proposed in the field of neurology. When the training gradient decreases, Softplus has faster convergence rate than the traditional Sigmoid and other saturated nonlinear functions, which improves the training speed and guarantees the predictive performance of the network meanwhile. For an input x, its corresponding output transformed with Softplus is $log(1 + e^x)$.

3.2. Two-stream UDN based on Improved UDN

The key to pedestrian detection is to find the image area containing human body. However, in actual pedestrian detection scenario, some columnar objects such as trees and telephone poles are often misjudged as pedestrians, as shown in Figure 5.

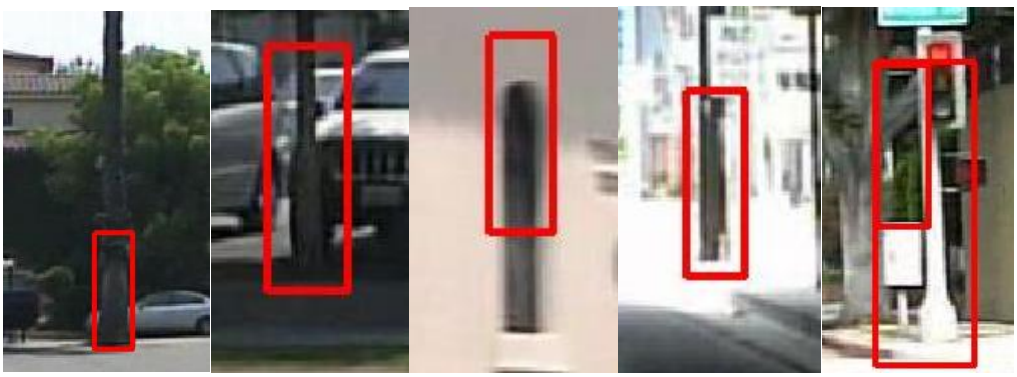


Figure 5. Examples of misjudged images with improved UDN

Due to the relatively minor change of posture of pedestrian's head, the region including head has better invariance and discrimination from the upper part of columnar objects such as trees and telephone poles. Therefore, we construct another branch of improved UDN to accept upper 1/3 pedestrian image as input.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2020 doi:10.20944/preprints202001.0029.v1

Peer-reviewed version available at Appl. Sci. 2020, 10, 1866; doi:10.3390/app10051866

This paper proposes a two-stream UDN network shown in Figure 6. The two-stream UDN consists of two parallel branches, the Global network and the Local network. The Global network is an improved UDN described in 3.1, details of each layer of the Local network are as following:

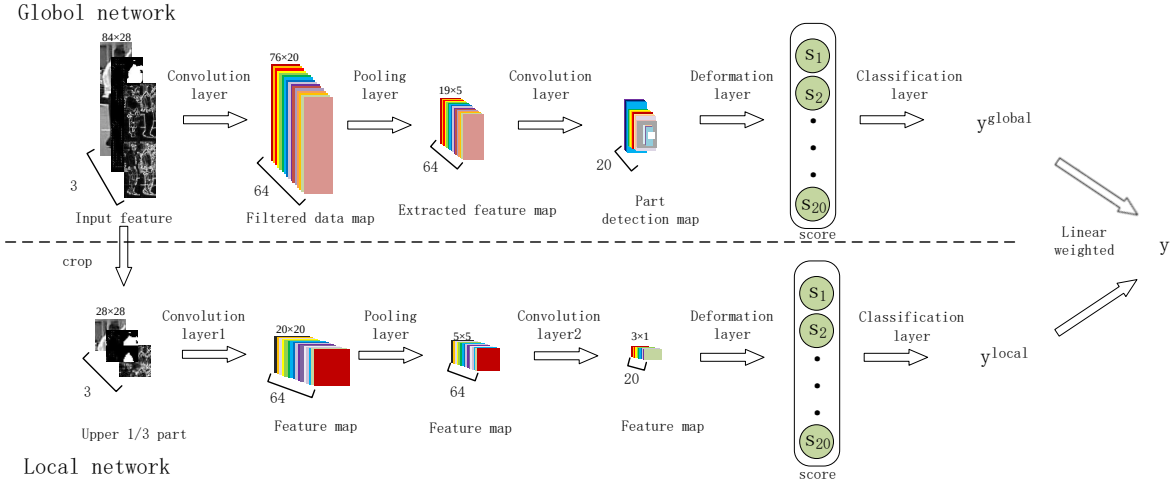


Figure 6. The two-stream UDN

1) Convolution layer 1: The input of the first convolution layer has three components $X_i^{local} \in R^{28\times28}, i = 1,2,3$, which are upper 1/3 part region of the original input image corresponding to the lower input channels in Figure 6. There are 64 convolution kernels $\{W_j^{local} | W_j^{local} \in R^{9\times9\times3}, \mathrm{j} = 1, \ldots, 64\}$ are utilized for extracting features. With each convolution kernel W_j^{local}, the feature map $A_j^{local} \in R^{20\times20}$ is computed as follows:

$$A_j^{local} = Softplus(b_j^{local} + \sum_{i=1}^{3}[W_j^{local}]_{:,:,i} * X_i^{local}) \quad (4)$$

where $[W_j^{local}]_{:,:,i}$ represents the i^{th} slice of three-order tensor W_j^{local}, $*$ represents convolution operator, b_j^{local} represents bias parameters for local network which is obtained by random initialization. Before training, W_j^{local} is initialized with Gabor filter.

2) Pooling layer: Calculate the average value of pixels in each 4 × 4 neighborhood on each feature map A_k^{local} and obtain extracted feature maps $\{B_k^{local} | B_k^{local} \in R^{5\times5}, \mathrm{k} = 1, \ldots, 64\}$, which is computed as follows:

$$B_k^{local} = AveragePool(A_k^{local}, 4, 4) \quad (5)$$

where k represents the number of the extracted feature maps.

3) Convolution layer 2: The second convolution layer has 20 filters $F_n^{local} \in R^{3\times5\times64}, n = 1, \ldots, 20$. Since the input of the local network is the upper 1/3 region of the image, there is no need to convolve in parts like the network based on the global input. The local feature map of the second Convolution layer C_n^{local} is computed as follows:

$$C_n^{local} = Softplus(u_n^{local} + \sum_{m=1}^{64}[F_n^{local}]_{:,:,m} * B_m^{local}) \quad (6)$$

where n represents the number of the filtered data maps, $[F_n^{local}]_{:,:,m}$ represents the m^{th} slice of three-order tensor F_n^{local}, $*$ represents convolution operator, u_n^{local} represents bias parameters obtained by random initialization. Before training, F_n^{local} uses Gabor filter for initialization.

4) Deformation layer: The deformation layer is same with the global network which reduces the number of parameters and extracts head-shoulder feature scores s_p^{local}, $\{s_p^{local} | p = 1, \ldots, 20\}$.

5) Classification layer: The visibility reasoning network [14] is used to estimate the label y^{local} by learning the head-shoulder feature scores s_p^{local}, which represents whether the upper 1/3 part contains a pedestrian head-shoulder or not.

With the above Global network and the Local network, the two-stream UDN gives final prediction for input image with (7) as final output:

$$y = \beta y^{global} + (1 - \beta) y^{local} \quad (7)$$

Where the value of β is set by experience.

3.3. Training tricks of Two-stream UDN

The Global network and the Local network of the two-stream UDN network are first trained separately, and each subnetwork adopts following variance loss in equation (8):

$$J(W,b,x,y) = \frac{1}{m}\sum_{i=1}^{m}(\frac{1}{2}||h_{W,b}(x^{(i)}) - y^{(i)}||^2) \tag{8}$$

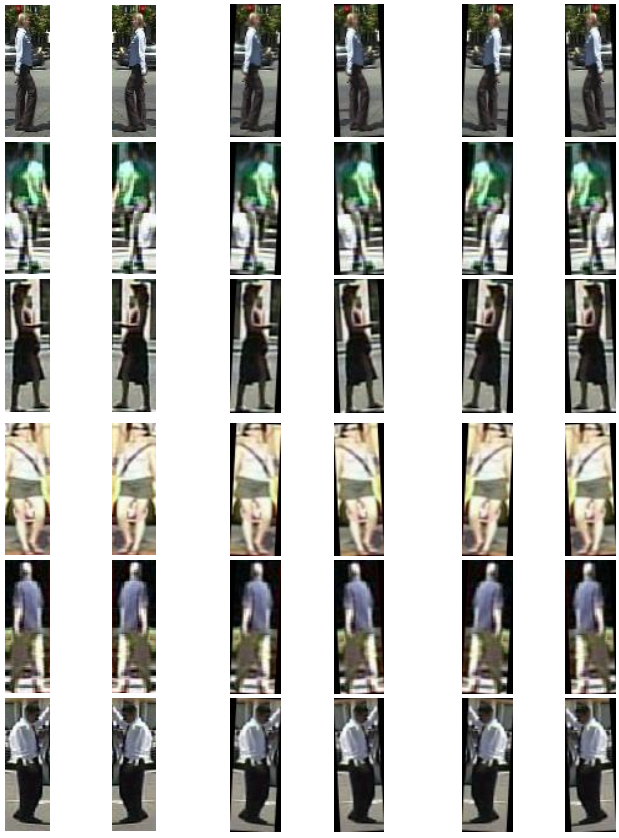
where y represents the ground truth label of the image, and x represents the input of the network, h represents the prediction label of the input x, m represents the number of total training samples. After the two sub-networks are pre-trained independently, the Global network $N_G(W_1, b_1)$ and the Local network $N_L(W_2, b_2)$ are fixed temporally.

Then jointly training the two sub-networks with the loss function defined in equation (9) to update parameters of the global network and the local network, which are (W_1, b_1) and (W_2, b_2).

$$J(W_1, b_1, W_2, b_2; x, y) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|\beta h_{W_1,b_1}(x^{(i)}) + (1-\beta)h_{W_2,b_2}(x^{(i)}) - y^{(i)}\right\|^2\right)\right] \tag{9}$$

During the joint training, the two sub-networks are updated alternately due to the differences in network input sizes and parameter volumes between the global and local networks.

Since pedestrian images are cut from street view videos captured by in-vehicle cameras, number of positive samples is too small. In order to improve generalization ability of the proposed method, it is necessary to expand positive samples. This paper augments data by applying mirror transformation, clockwise rotation and counter clockwise rotation to positive samples. The rotation angle is three degree and bilinear interpolation is adopted during rotation. Figure 7 shows some examples of mirror transformation and angular rotation.



(a) (b) (c) (d) (e) (f)

Figure 7. The examples of positive sample augmentation. (a) origin images; (b) images after a mirror flip; (c) images with a clockwise rotation of three degree; (d) images with a counter clockwise rotation of three degree; (e) mirrored images with a clockwise rotation of three degree; (f) mirrored images with a counter clockwise rotation of three degree.

4. Experimental results

Training and testing images used in the experiments are same with Wanli Ouyang et al. [21], which are preprocessed from the Caltech pedestrian dataset using HOG + CSS + SVM. The original image size is 108×36. Before fed into networks, images are cropped to 84×28. Positive sample is an image including a pedestrian, and negative sample is an image without pedestrian.

During network training, the input samples are randomly selected. 60 samples are used in each batch, including 50 negative samples and 10 positive samples. The ratio of positive and negative samples is 1: 5. The reason for choosing this ratio is to make it closer to what happens in real life scenarios faced by pedestrian detection system. With the augment method in 3.3, approximately 60,000 negative samples and 12,000 positive samples are available for the INRIA, and approximately 60,000 negative samples and 24,000 positive samples are available for the Caltech.

The Log-average Miss Rate (LAMR) is used as performance evaluation for evaluating different pedestrian detection methods, for which lower value means better performance.

In the process of parameter training, first fix the learning rate to 0.05. When the LAMR does not decrease, subtract 0.005 from the learning rate and continue until the learning rate is less than 0.02 and the LAMR no longer decreases. The network reaches optimal solution.

4.1. Improved UDN

In this section, the improved UDN model is evaluated. The input images are preprocessed following the rules mentioned in Section 3.1.

4.1.1. Comparison of activation functions

Using Sigmoid and Softplus as activation function separately, starting with same network initialization, after trained 5 epochs, the LAMR of 42.63% and 43.19% are achieved, and time cost of each epoch is shown in Table 1. It can be seen from Table 1, when the activation function is Softplus, the parameter training speed increases significantly without large performance degradation.

Therefore, using Softplus as activation function can significantly improve calculation speed without sacrificing detection accuracy. In following experiments, activation function adopts Softplus.

Table 1. Training time while using Sigmoid or Softplus as activation function

	1th	2th	3th	4th	5th	Average
Sigmod	5249s	5298s	5379s	4013s	4029s	4793s
Softplus	3740s	3658s	3660s	3574s	3535s	3633s

4.1.2. Comparison with hand-crafted feature based models

The first experiment trained the improved UDN with the set00-set05 of the Caltech dataset, and tested all models on the set06-set10. The second experiment trained the improved UDN with the INRIA database and tested all models on the ETH database. As seen in Table 2, improved UDN is superior to other models based on manual designed features.

Table 2. LAMR of improved UDN and hand-crafted feature based models.

	VJ[29]	HOG[8]	HOG+LBP[31]	ChnFtrs[10]	ACF[32]	Improved UDN
Caltech	94.73%	68.46%	67.77%	56.34%	51.04%	38.73%
ETH	89.89%	64.23%	55.00%	57.00%	50.04%	45.04%

4.1.3. Analysis of multiple channel inputs

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2020 doi:10.20944/preprints202001.0029.v1

Peer-reviewed version available at Appl. Sci. 2020, 10, 1866; doi:10.3390/app10051866

Figure 8 shows the experimental results of investigating the influence of different channel combinations introduced in Section 3.1. When the input data is the original RGB three channel, the LAMR is 44.61%. When the input data is the first channel, the LAMR is 46.13%. When the input data includes both the first and the second channels, the LAMR is reduced by 4.77%. When the input data includes all the three channels, the LAMR is reduced furtherly by 2.63%.

The comparison results show original RGB image has the highest miss rate, which is lower than using the first channel lonely. Combining more feature descriptions with the first channel, much lower miss rate is obtained. Thus, multi-channel feature input method can enhance the feature learning ability of network, and improve the recognition performance of network.

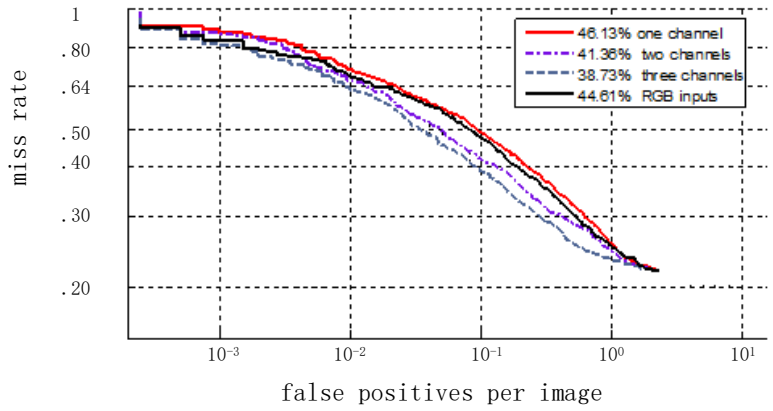


Figure 8. Results of various input channels with the Caltech-Test dataset

4.2. Two-stream UDN

In this section, experimental verification of the two-stream UDN for pedestrian detection is performed.

4.2.1. Effectiveness of adding local network

Table 3 shows the LAMR of the two-stream UDN and its two sub-networks on the Caltech and ETH datasets. As can be seen from Table 3, the LAMR of the two-stream UDN network for pedestrian detection on the Caltech dataset is 37.66%.

Compared to the Global network, the two-stream UDN improved the miss rate by 1.07%. If only considering local features, the detection result is 58.15%, because the local network is only sensitive to partial region, and it is easy to misjudge round objects as persons. Combing the global network and local network together, results of the two-stream UDN network are better than each single network. Similar trend shows on the ETH data set, which means joint training of global feature and local features are effective for pedestrian detection.

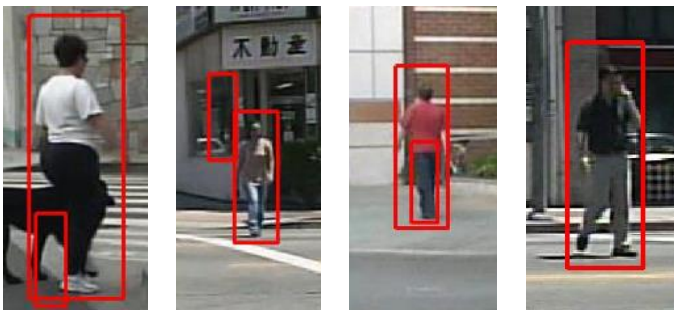
Table 3. LAMR of different networks.

	Global network	**Local network**	**Two-stream UDN**
Caltech	38.73%	58.15%	37.66%
ETH	45.04%	68.82%	44.58%

Scores of the two-stream UDN network and the Global network for positive samples are shown in Figure 9. In Figure 9, (a) represents score obtained by the global network, and (b) represents score obtained by the two-stream UDN. The results show that the local network increase the confidence of the positive samples, which makes judgments for the positive samples more accurate.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2020 doi:10.20944/preprints202001.0029.v1

Peer-reviewed version available at Appl. Sci. 2020, 10, 1866; doi:10.3390/app10051866



(a) 0.4910 (b) 0.4964 | (a) 0.4857 (b) 0.4898 | (a) 0.4022 (b) 0.4864 | (a) 0.4732 (b) 0.4898

Figure 9. Scores for positive samples in Caltech with the global network and the improved UDN

Scores of the two-stream UDN network and the Global network for negative samples are shown in Figure 10. In Figure 10, (a) represents core obtained by the global feature, and (b) represents score obtained by the two-stream UDN. Aim of the local network is to find features that match the head and shoulders of pedestrians. Thus, negative samples whose upper 1/3 region do not have head and shoulder features will obtain lower scores than that of global network. This illustrates that the extraction of head and shoulder features can be effectively enhanced by adding local network. Thus, the proposed two-stream UDN structure can improve pedestrian detection result.

(a) 0.4674 (b) 0.3751 | (a) 0.4727 (b) 0.3863 | (a) 0.4858 (b) 0.2053 | (a) 0.4746 (b) 0.3896

Figure 10. Scores for false samples in Caltech with the global network and the improved UDN

4.2.2. Comparison with other shallow nets

Table 4 shows comparison of the two-stream UDN and other shallow nets on Caltech and ETH. All the compared networks have two or three layers, and all their input features are hand crafted features, such as HOG, gradient or color self similarity (CSS).

Comparing with other shallow nets, the improved UDN gets the lowest score on the Caltech and the second lowest score on the ETH. Comparing with the UDN, the improved UDN improves more obviously on the Caltech. A possible reason is that images in ETH and INRIA have higher resolution, so the global network can extract better features on the upper region, so the local network benefits less. The results indicate that the two-stream UDN works better when the captured image has low resolution.

Table 4. LAMR of Two-stream UDN and shallow networks.

	ConvNet[19]	DBN-ISO[14]	DBN-Mut[20]	UDN[21]	SDN[22]	MultiSDP[34]	Two-stream UDN
Caltech	77.20%	53.29%	48.22%	39.32%	37.87%	45%	37.66%
ETH	50.27%	47.01%	41.07%	45.32%	40.63%	48%	44.58%

5. Conclusions

This paper proposed a two-stream UDN for pedestrian detection, which includes two subnetworks trained jointly and used for judgement independently. The two subnetworks give prediction based on the overall or the upper 1/3 region of the bounding box, accept the V channel in HSV space, HOG feature map, Sobel feature map, and GrabCut feature map as input.

Experiments show that the proposed structure can effectively reduce the score of non-pedestrian objects such as trees and telephone poles, and improve the detection effect of pedestrians.

Author Contributions: Conceptualization, W.W. and L.W.; methodology, L.W.; software, W.W. and X.G.; validation, W.W., L.W. and X.G.; formal analysis, W.W.; investigation, X.G.; resources, B.Y.; data curation, X.G.; writing—original draft preparation, W.W.; writing—review and editing, W.W., L.W. and J.L.; supervision, B.Y.; project administration, B.Y.; funding acquisition, B.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by National Natural Science Foundation of China (61876012, 61772049), and Beijing Natural Science Foundation(4202003).

Conflicts of Interest: The authors declare there is no conflicts of interest regarding the publication of this paper.

References

1. World Health Organization. Global Status Report on Road Safety 2018. http://www.who.int/violence_injury_prevention/road_safety_status/2018/en/ (accessed on 20/12/2019).
2. Gavrila, D. M.; Giebel, J.; Munder, S. Vision-based Pedestrian Detection: The Protector System. Intelligent Vehicles Symposium, Parma, Italy, 2004:13-18.
3. Gavrila, D. M. Pedestrian Detection from a Moving Vehicle. Proceedings of the 6th European Conference on Computer Vision-Part II. London, UK: Springer-Verlag, 2000:37-49.
4. Shashua, A.; Gdalyahu, Y.; Hayun, G. Pedestrian Detection for Driving Assistance Systems: Single-frame Classification and System Level Performance. Intelligent Vehicles Symposium, Parma, Italy, 2004:1-6.
5. Wu, B.; Nevatia, R. Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. *International Journal of Computer Vision*, 2007, 75(2):247-266.
6. Viola, P.; Jones, M.J.; Snow, D. Detecting Pedestrians Using Patterns of Motion and Appearance. *International Journal of Computer Vision*, 2013, 63(2):153–161.
7. Lowe, D.G. Distinctive Image Features from Scale Invariant Key Points. *International Journal of Computer Vision*, 2004, 60(2):91-110.
8. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, 1:886-893.
9. Walk, S.; Majer, N.; Schindler, K. et al. New Features and Insights for Pedestrian Detection. IEEE Conference on Computer Vision and Pattern Recognition, 2010:1030-1037.
10. Dollár, P.; Tu, Z.; Perona, P. Integral Channel Features. The British Machine Vision Conference, 2009, London, England.
11. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D. et al. Object Detection with Discriminatively Trained Part-based Models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2010, 32(9):1627-1645.
12. Felzenszwalb, P.F.; Girshick, R.B; McAllester, D. et al. Cascade Object Detection with Deformable Part Models. IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2010:2241-2248.
13. Fischler, M.A.; Elschlager, R.A. The representation and matching of pictorial structures. *IEEE Transactions on computers*, (1), 67-92.
14. Ouyang, W.; Wang, X. A discriminative Deep Model for Pedestrian Detection with Occlusion Handling. IEEE Conference on Computer Vision and Pattern Recognition, 2012:3258-3265.
15. Dollár, P.; Wojek, C.; Schiele, B. et al. Pedestrian Detection: A Benchmark. IEEE Conference on Computer Vision and Pattern Recognition, 2009:304-311.
16. Zeng, X.; Ouyang, W.; Wang, M. et al. Deep Learning of Scene-specific Classifier for Pedestrian Detection. European Conference on Computer Vision. Springer, Cham, 2014:472-487.
17. Cortes, C.; Vapnik, V. Support-Vector Networks. *Machine Learning*, 1995, 20(3):273-297.
18. Benenson, R.; Omran, M.; Hosang, J. et al. Ten Years of Pedestrian Detection, What Have We Learned? European Conference on Computer Vision. Springer, Cham, 2014, 8926:613-627.

19. Sermanet, P.; Kavukcuoglu, K.; Chintala, S. et al. Pedestrian Detection with Unsupervised Multi-stage Feature Learning. IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2013:3626 - 3633.
20. Ouyang, W.; Zeng, X.; Wang, X. Modeling Mutual Visibility Relationship in Pedestrian Detection. IEEE Conference on Computer Vision and Pattern Recognition, 2013:3222-3229.
21. Ouyang, W.; Wang, X. Joint Deep Learning for Pedestrian Detection. IEEE International Conference on Computer Vision, 2013:2056-2063.
22. Luo, P.; Tian, Y.; Wang, X. et al. Switchable Deep Network for Pedestrian Detection. IEEE Conference on Computer Vision and Pattern Recognition, 2014:899-906.
23. Ouyang, W.; Zhou, H.; Li, H. et al. Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection. *IEEE transactions on pattern analysis and machine intelligence*, 2017, 40(8): 1874-1887.
24. Brazil, G.; Yin, X.; Liu, X. Illuminating pedestrians via simultaneous detection & segmentation. Proceedings of the IEEE International Conference on Computer Vision. 2017: 4950-4959.
25. Wang, S.; Cheng J.; Liu, H. et al. PCN: Part and context information for pedestrian detection with CNNs. BMVC 2017, London, UK
26. Zhang, S.; Yang, J.; Schiele B. Occluded pedestrian detection through guided attention in CNNs. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 6995-7003.
27. Zhang, X.; Cheng, L.; Li, B. et al. Too far to see? Not really!—pedestrian detection with scale-aware localization policy. *IEEE transactions on image processing*, 2018, 27(8): 3703-3715.
28. Song, T.; Sun, L.; Xie, D. et al. Small-scale pedestrian detection based on somatic topology localization and temporal feature aggregation. In ECCV. Springer, 2018.
29. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. Journal of Machine Learning Research, 2011, 15:315-323.
30. Wang, X.; Han, T.X.; Yan, S. An HOG-LBP human detector with partial occlusion handling. 2009 IEEE 12th international conference on computer vision. IEEE, 2009: 32-39.
31. Dollár, P.; Appel, R.; Belongie, S. et al. Fast feature pyramids for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2014, 36(8): 1532-1545.
32. Zeng, X.; Ouyang, W.; Wang, X. Multi-stage contextual deep learning for pedestrian detection. Proceedings of the IEEE International Conference on Computer Vision. 2013: 121-128.