Image and Feature Space Based Domain Adaptation for Vehicle Detection

Ying Tian[1, *], Libing Wang[1], Hexin Gu[2] and Lin Fan[3]

Abstract: The application of deep learning in the field of object detection has experienced much progress. However, due to the domain shift problem, applying an off-the-shelf detector to another domain leads to a significant performance drop. A large number of ground truth labels are required when using another domain to train models, demanding a large amount of human and financial resources. In order to avoid excessive resource requirements and performance drop caused by domain shift, this paper proposes a new domain adaptive approach to cross-domain vehicle detection. Our approach improves the cross-domain vehicle detection model from image space and feature space. We employ objectives of the generative adversarial network and cycle consistency loss for image style transfer in image space. For feature space, we align feature distributions between the source domain and the target domain to improve the detection accuracy. Experiments are carried out using the method with two different datasets, proving that this technique effectively improves the accuracy of vehicle detection in the target domain.

Keywords: Deep learning, cross-domain, vehicle detection.

1 Introduction

Vehicle detection is a fundamental problem in computer vision. Due to the research progress of convolutional neural network (CNN) in recent years, the vehicle detection method based on CNN has made significant achievements. Researchers [Song, Liang, Li et al. (2019)] have proposed a vision-based vehicle detection and counting system to detect vehicles. This segmentation method can provide higher vehicle detection accuracy, especially for the detection of small vehicles. Other scholars [Zhang and Zhu (2020)] have conducted vehicle detection using fast image registration and You Only Look Once version 3 (YOLOv3) network. This method can achieve satisfactory and competitive moving

[1] School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China.

[2] School of Mechanical Engineering and Automation, University of Science and Technology Liaoning, Anshan, 114051, China.

[3] Faculty of Business, Economics & Law, The University of Queensland, Brisbane, QLD 4072, Australia.

[*] Corresponding Author: Ying Tian. Email: astianying@126.com.

Received: 06 May 2020; Accepted: 05 June 2020.

vehicle detection results. However, variability issues that commonly exist in object detection, such as shooting angles, the scene environment, and image quality, will lead to a domain shift between the source domain and the target domain. As shown in Fig. 1 below, Kitti [Geiger, Lenz, Stiller et al. (2013)] and Cityscapes [Cordts, Omran, Ramos et al. (2016)] datasets used in the field of autonomous driving show obvious domain shifts.

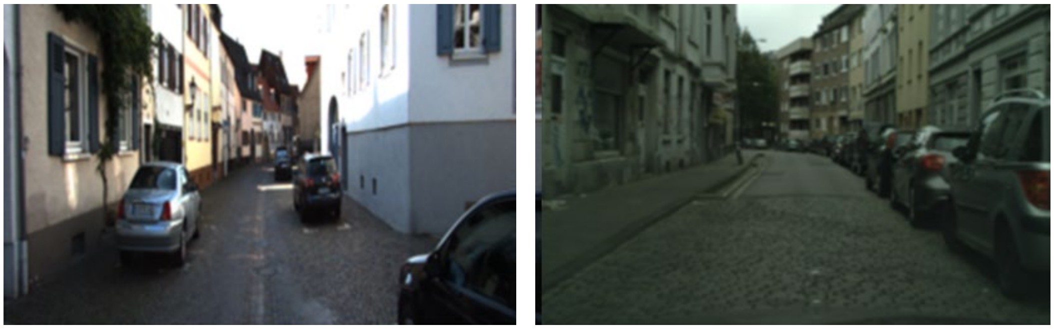

Figure 1: Domain shifts in Kitti and Cityscapes datasets

The left example image shows the Kitti dataset, and the right one shows Cityscapes. Although there are two datasets covering the urban scenes, images in those datasets vary in different external environments and lighting.

Such domain shifts can lead to a significant performance drop. In addition, annotating the data requires significant human and financial resources. Domain adaptation is one method to solve these problems.

The focus of this work is the problem of unsupervised domain adaptation, in which the source domain has complete data annotations, but the target domain has no data annotations, and the data distribution of the source domain and the target domain is different within the same detection task. A new domain adaptation method for vehicle detection in then proposed, and the detection accuracy of the model under different datasets is improved through testing and verification among different datasets.

2 Literature review

2.1 Object detection

In recent years, CNN [Wu, Liu and Liu (2019)] has been widely used in the field of object detection, and the region-based CNN (RCNN) method has advanced significantly. Object detection methods based on CNN can be divided into two categories: the two-stage method and the single-stage method. The two-stage method is based on RCNN and mainly extracts the bounding box from images, then trains a network to recognize each region of interest (ROI) [Girshick, Donahue, Darrell et al. (2014)]. The method of sharing convolutional feature graph for all ROI was proposed to expand RCNN. Region proposal network (RPN) was then employed in Fast RCNN [Girshick (2015)], which provided high accuracy but low speed. Single-stage methods include YOLO [Redmon, Divvala, Girshick et al. (2016); Redmon and Farhadi (2018)] and SSD [Liu, Anguelov, Erhan et al. (2016); Fu, Liu, Ranga et al. (2017)]. The advantage of this type of method is that there is no need to extract a bounding box, and it is high speed but low accuracy.

2.2 Domain adaptation

Domain adaptation is a special transfer learning method as it attempts to solve the learning problem in which the target domain is without annotations while the source domain has complete annotations. It has been studied for a long time in the field of computer vision. Pan et al. [Pan, Tsang, Kwok et al. (2011)] found that Transfer Component Analysis (TCA), a kernel method based on maximum mean discrepancy (MMD) [Zhang, Liu, Luo et al. (2018)], was able to learn better feature representation across domains. Based on MMD, Long et al. [Long, Wang, Ding et al. (2013); Long, Cao, Wang, et al. (2015); Long; Zhu, Wang et al. (2016)] found that hidden network features in a reproducing kernel Hilbert space and explicitly measures the difference between the two domains using MMD and its variants. Sun et al. [Sun, Feng and Saenko (2016)] attempted to minimize the domain shift by aligning the second-order statistics of feature distribution between the source and target domain.

2.3 Image style transfer

In recent years, deep learning technology based on convolutional neural network has been popularized, and research into image style transfer has also flourished. By employing a convolutional neural network that can effectively extract image features, researchers [Gatys, Ecker and Bethge (2016)] proposed an automatic method for image style transfer. They believed that the image of style transfer should contain both content features and style features of the image. Gatys et al. [Gatys, Bethge, Hertzmann et al. (2016)] further presented an image style transfer method that could retain the color of the original image, so that the final generated image could obtain the texture features of the image while still retaining the color distribution. Johnson et al. [Johnson, Alahi and Li (2016)] proposed an image style transfer method based on feedforward network, and divided the transfer method into two different networks: image conversion network and loss function network. Dumoulin et al. [Dumoulin, Shlens, Kudlur et al. (2016)] set up a transfer network containing multiple style diagrams, pointing out that some parameters of various styles in the network could be shared.

The recently proposed CycleGAN model [Chen, Li, Sakaridis et al. (2018)] is a promising method for unpaired image style transfer. It has yielded convincing results, such as converting aerial images to Google maps or Monet paintings to images. Using this method, cycle consistency loss is employed to regularize the generative model and preserve the transferred image's structural information. However, this approach only ensures that a region is occupied by an object before image style transfer and after image style transfer. The semantics of pixels are not guaranteed to be consistent with this, only cycle consistency loss [Li, Liang, Jia et al. (2018); Bousmalis, Silberman, Dohan et al. (2017); Hoffman, Tzeng, Park et al. (2017); Zhu, Park, Isola et al. (2017)] proposed the use of semantic labeled images as additional signals to regularize the generative models of CycleGAN to generate the same segmentation images. However, this method requires the training of additional segmented networks, which can slow down the whole training process. For certain tasks, such as object detection, the same consideration is not required for all pixels.

3 Proposed method

In this paper, we propose a vehicle detection method based on Faster RCNN. Faster RCNN model architecture mainly includes three main parts: a convolutional layer of shared extracted feature map, RPN, and ROI. In addition, the feature maps are extracted from the input images through the convolutional neural network, generally employing VGG16 or ResNet-50. In this paper, the convolutional layer of VGG16 is used as a feature extractor.

To extract the characteristics of the figure, the RPN network generates a bounding box and conducts preliminary object detection to the bounding box, which feature map with the results back to the last convolutional layer diagram and unify the ROI size. ROI pooling will then input the region to the full connection layer, which will return to the position of the target domain and use ROI to category forecast. The training loss is composed of the loss of the RPN and the loss of the ROI, and is defined as:

$$L_d = L_{rpn} + L_{roi} \tag{1}$$

Both the training loss of the RPN and ROI has two loss terms: One is used for classification, that is, to predict how accurate the probability is, and the other is a regression loss on the box coordinates for better localization. The loss function can be written as:

$$L(\{p_i\},\{t_i\}) = L_A + L_B \tag{2}$$

where L_A is the loss of classification, and L_B is the loss of regression. L_A is formulated as:

$$L_A = \frac{1}{N_{cls}} \sum_{\mathrm{i}} L_{\mathrm{cls}}(p_i, p_i^*) \tag{3}$$

The anchor generated by the RPN module is only divided into the foreground and background. The label of the foreground is 1, and the label of the background is 0. During the training of RPN, 256 anchors are selected, which are the N_{cls} in Eq. (3).

where P_i is the probability of anchor prediction as the target, with the ground-truth label $p_i^* = \begin{cases} 0 & \textit{negative} \\ 1 & \textit{positive} \end{cases}$. $L_{\mathrm{cls}}(P_i, P_i^*)$ is the logarithmic loss for two categories (target or non-target) and L_{cls} is formulated as:

$$L_{\mathrm{cls}}(p_i, p_i^*) = -\log[p_i^* p_i + (1-p_i^*)(1-p_i)] \tag{4}$$

The loss of regression L_B is defined as:

$$L_B = \lambda \frac{1}{N_{\mathrm{reg}}} \sum_{\mathrm{i}} \mathrm{p}_{\mathrm{i}}^* L_{\mathrm{reg}}(t_i, t_i^*) \tag{5}$$

where $t_i=\{t_x, t_y, t_w, t_h\}$ is a vector representing the offset predicted by the anchor during the RPN training phase. t_i^* is the same vector as the t_i dimension, indicating that the offset of the anchor is relative to ground-truth during the RPN training phase, and λ is used to weigh L_{cls} and L_{reg} with a default value of 10. L_{reg} can be written as:

$$L_{\text{reg}}(t,t_i^*) = R(t_i - t_i*) \tag{6}$$

where R is the $smooth_{L1}$ loss function, which is defined as:

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & if|\text{x}|<1 \\ |x|-0.5 & otherwise \end{cases} \tag{7}$$

3.1 Image space based domain adaptation

The CycleGAN model is used for image style transfer in the image space. Specifically, the image style is transferred from the target domain dataset Kitti to the source domain dataset Cityscapes. The images generated by the image style transfer are labeled with the corresponding images in the source domain dataset Cityscapes. The image generated by the image style transfer is shown in Fig. 2, and contains the image information from both datasets.

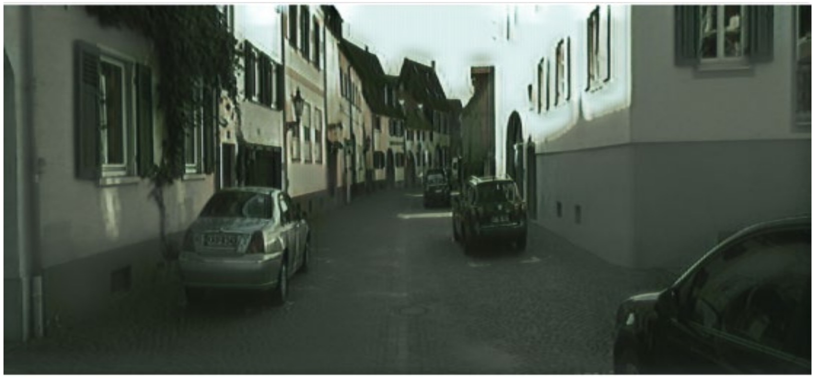

Figure 2: Generated image by image style transfer

As can be seen from Fig. 2, the generated image using the CycleGAN model contains the external environment and lighting of the target domain dataset Cityscapes. Specifically, the task of CycleGAN is to transfer the image style from source domain A to target domain B. In order to train the unpairing data, two GAN [Liu and Tuzel (2016); Zhang and Dana (2017); Mao, Li, Xie et al. (2017)] loss functions are calculated. The training of the whole network is carried out according to reference [Girshick, Donahue, Darrell et al. (2014)]. The whole cycle consistency loss is formulated in Eq. (8):

$$L_{cyc}(f_{AB}, f_{BA}) = E_{a\sim A}[\| f_{BA}(f_{AB}(a)) - a \|_1] + E_{b\sim B}[\| f_{AB}(f_{BA}(b)) - b \|_1] \tag{8}$$

The items in Eq. (8) are shown in Fig. 3. The meaning of the first item of this loss function is that the image in source domain A should be similar to the original image a after the generator G_A is used to generate the $f_{AB}(a)$, and then $f_{AB}(a)$ changes from the generator G_A to the $f_{BA}(f_{BA}(a))$. The second item of the loss function is that image b in target domain B generates $f_{BA}(b)$ after being acted on by generator G_B, and then $f_{BA}(b)$ changes from generator G_A to $f_{BA}(b)$, which should be similar to the original image b. After adding constraints in this way, a certain relationship is established between unmatched data. In Fig. 3, we show the domain adaptation based on image space through first converting the source domain A to the target domain B, and then using it to train the vehicle detection network.

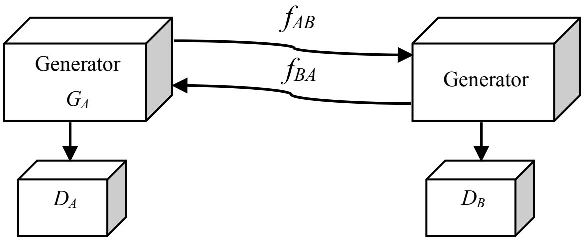

Figure 3: Architecture of CycleGAN

3.2 Feature space based domain adaptation

In feature space, for the object detection model Faster RCNN, feature-based domain adaptation is the domain adaptation of feature map extracted by convolutional neural network. In this paper, the domain classifier is added to the two levels of the low feature map and the high feature map, and used to classify different domains. The added domain classifier is shown in the last phase of Fig. 4.

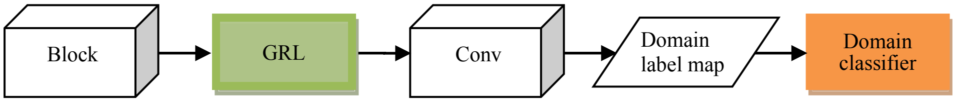

Figure 4: The added block of domain classifier (FS module)

In Fig. 4, GRL is the gradient reversal layer [Ganin and Lempitsky (2015)] and Conv is the convolutional layer. The added block of domain classifier is referred to as FS module. The convolutional feature map corresponds to the input of the added block, and the function of the domain classifier is to determine the domain of each image block.

With cross-entropy loss, the calculation formula of domain classifier loss based on feature graph is as follows:

$$L_{high}, L_{low} = \sum_{i,u,v}[D_i \log P_i^{(u,v)} + (1-D_i)\log(1-p_i^{(u,v)})] \tag{9}$$

where I_i shows that the training image has i images, so that D_i represents the domain of the training image. D_i=0 represents the source domain and D_i=1 represents the target domain. The output results of domain classifier based on feature map are recorded as $P_i^{(u,v)}$.

3.3 Method description

The main idea of domain adaptation based on feature map is to extract features with domain invariance. The worse the classification effect of domain classifier, the better the common representation of the feature map to the source domain and target domain. Therefore, the smaller the adaptation loss function of the above domain, the better the adaptability of the model in the target domain. Thus, we need to optimize the parameters of domain classifier to minimize the domain adaptation loss based on feature map and optimize the parameters of the whole convolutional network to maximize the loss. In the process of the practical experiment, referring to the practice in reference [Ganin and

Lempitsky (2015)], the method of gradient drop is employed in training, and a gradient reversal layer (GRL) is introduced at the same time. When passing through the GRL layer, the gradient reversed the direction to optimize the basic network. The function of this layer is to satisfy both the parameter optimization requirements of Faster RCNN and the domain adaptation components. The architecture of the vehicle detection model is shown in Fig. 5.

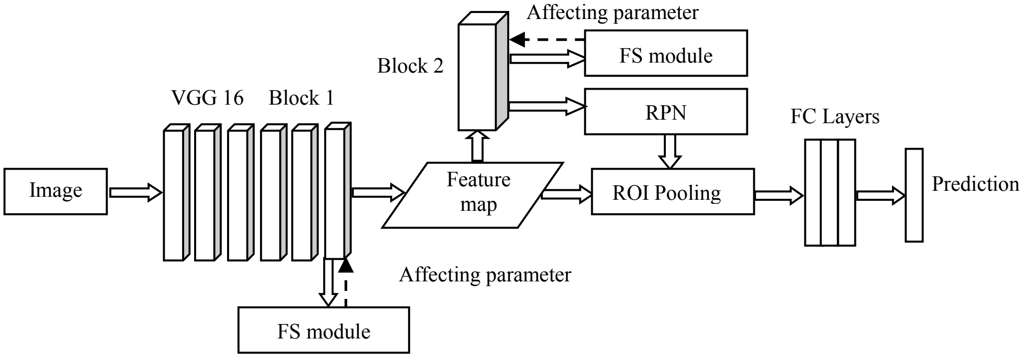

Figure 5: Architecture of the vehicle detection model

As can be seen from Fig. 5, detection network is enhanced with domain classification for multi-feature level based adversarial training. FS module is implemented on the block 1 and block 2 of the network, which affects the parameters of the convolutional layer.

Two novel components are introduced in our feature space based domain adaptation. The domain classifier is added after feature map and the last convolutional layer. It is suggested that the final training loss of the network is the sum of all parts, which can be written as:

$$L = L_d + \lambda(L_{high} + L_{low}) \tag{10}$$

where λ is a trade-off parameter to balance the Faster RCNN loss and our newly added domain adaptation classifiers. In the method process, the CycleGAN model of image style transfer is first carried out on the image space. Namely, the CycleGAN model is used to generate several images corresponding to the number of the Kitti dataset, and the generated dataset is assumed to be *A*. The data annotations in source domain dataset Cityscapes are then used in dataset *A*. When training the model, the corresponding images and annotations in source domain dataset Cityscapes and dataset *A* are used to obtain the pre-training model. The domain classifier is carried out in the feature space, that is, the domain classifier is trained using the target data. From the perspective of domain adaptation, the smaller the loss function of the domain classifier, the better the adaptability of the model to the target domain. After completing the three-part training, the model is saved, and the images of the target domain are used for testing.

In this paper, the CycleGAN model and the idea of domain classifier are applied to domain adaptive vehicle detection for the first time. In the past, CycleGAN model was mostly used for image style transfer. Previous works have largely only applied domain classifier to a single feature map. In contrast, it is applied to domain adaptive vehicle detection in our method in combination with feature-based domain adaptive vehicle detection network. We then apply domain classifier to two different feature maps. Namely, the idea of domain classifier is used for reference in the feature-based domain adaptive vehicle detection network.

4 Implementation

4.1 Dataset

Two datasets were used for the vehicle detection experiments: Kitti [Geiger, Lenz, Stiller et al. (2013)] and Cityscapes [Cordts, Omran, Ramos et al. (2016)]. The Cityscapes dataset is the image segmentation dataset used in autonomous driving and is generally employed to evaluate the performance of visual methods in semantic understanding of urban scenes. The Cityscapes dataset has 2975 training images and 500 images for validation.

The Kitti dataset was made by the Karlsruhe Institute of Technology in Germany and the Toyota Technological Institute in the United States. Toyota currently holds the largest global evaluation dataset for the computer vision method of autonomous driving. It contains urban, rural, and highway scenarios, including collections of real image data. In this paper, a total of 7481 labeled images were used for the experiment. Mutual domain adaptive experiments of Cityscapes and Kitti datasets were conducted to verify the effectiveness of the proposed method in solving domain adaptive problems caused by different scenes in vehicle detection.

4.2 Evaluating metrics

As this experiment only calculated the Average Precision (*AP*) for this category of vehicle, the effectiveness of the model on the vehicle detection task was measured. Generally speaking, the higher the *AP* is, the better the detection effect will be. Several concepts and definitions are first introduced here to describe the evaluation process. When evaluating the effect of target detection, the *AP* index was used as follows:

$$AP = \int_0^1 P(R)dR \tag{11}$$

where *P* is the precision rate, and *R* is the recall rate. The accuracy rate represents the proportion of true positive (*TP*) among the detected targets, that is, the proportion of detected vehicles that were the true targets of vehicles. The recall rate indicates the proportion of all positive samples in the entire test set that were correctly identified as positive samples, that is, the proportion of vehicles correctly detected in the real vehicle target. The formula is as follows:

$$P = \frac{TP}{TP + FP} \tag{12}$$

$$R = \frac{TP}{TP + FN} \tag{13}$$

Classifying the positive example correctly as a positive example means that the number of vehicles was correctly detected and denoted as *TP*. Classifying the positive example incorrectly as a negative example means that the vehicle target was not detected as the number of vehicles and denoted as *FN* (false negative). Classifying the negative example correctly as a negative example means that the target of non-vehicle was detected as the number of vehicles and denoted as *TN* (true negative). Classifying the negative example incorrectly as a positive example means the target of a vehicle was not detected as the number of vehicles and denoted as *FP* (false positive).

In the calculation of accuracy and recall, the coincidence rate between the bounding box and the ground truth box was set to *IOU* (intersection over union).

$$IOU = \frac{BOX_P \cap BOX_G}{BOX_P \cup BOX_G} \tag{14}$$

where BOX_P is the bounding box, BOX_G is the ground truth box, $\cap$ is the intersection area between the bounding box and the ground truth box, and $\cup$ is the combined area of the bounding box and the ground truth box.

4.3 Implementation details

Constrained by the GPU memory, we scaled the height of the image to 256 in the training stage and then cropped image patches with a size of 256×256 for the image space based on CycleGAN. The training images in the source data had complete annotations, while the test images in the target data had no labeling information. Both datasets were converted to VOC [Everingham, Eslami, Van Gool et al. (2015)] dataset format, and both focused on the category of vehicle. The initial parameters of the convolutional network were those of the pre-trained VGG16 network on ImageNet, and Caffe framework was adopted. Training was conducted a total of 50,000 times, the learning rate of the first 40,000 was 0.001, while that of the last 10,000 iterations gradually decreased. The weight of domain adaptation loss was set to 0.1.

4.4 Results and analysis

We evaluated our proposed domain adaptive model for vehicle detection in two different scenarios: 1) The source domain training images were 2975 images in Cityscapes, the target domain training images were 7481 images in Kitti, and the target domain test images were 7481 images in Kitti; 2) The source domain training images were 7481 images in Kitti, the target domain training images were 2975 images in Cityscapes, and the target domain test images were 500 images in Cityscapes.

Figs. 6-9 show the visualized performance comparison with the results of vehicle detection before and after adaptation. In Figs. 6 and 8, due to domain shift, vehicle detection results were not accurate enough to detect small and multiple vehicles. In Figs. 7 and 9, compared with the image of vehicle detection before adaptation, the result was accurate enough to detect small and multiple vehicles.

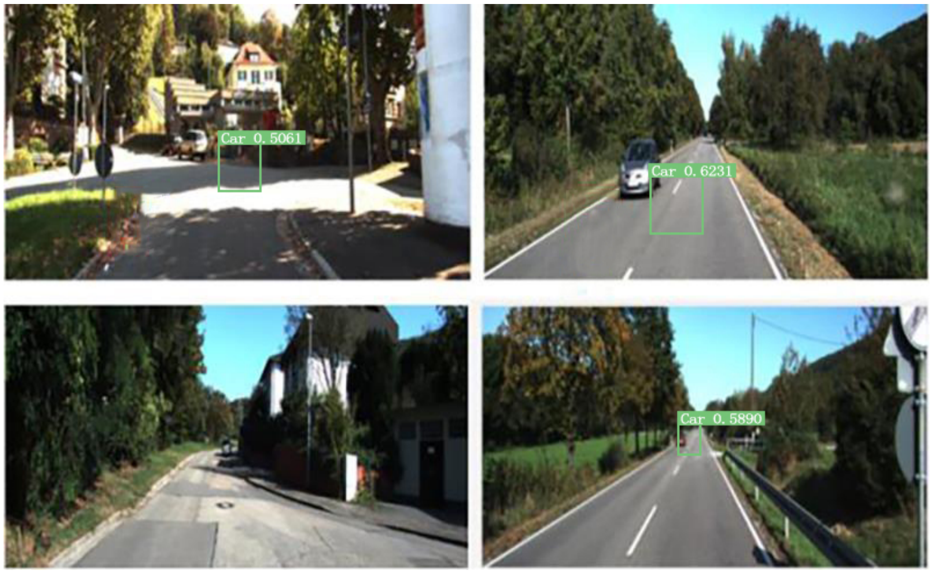

Figure 6: Example images of vehicle detection before adaptation in the first scenario

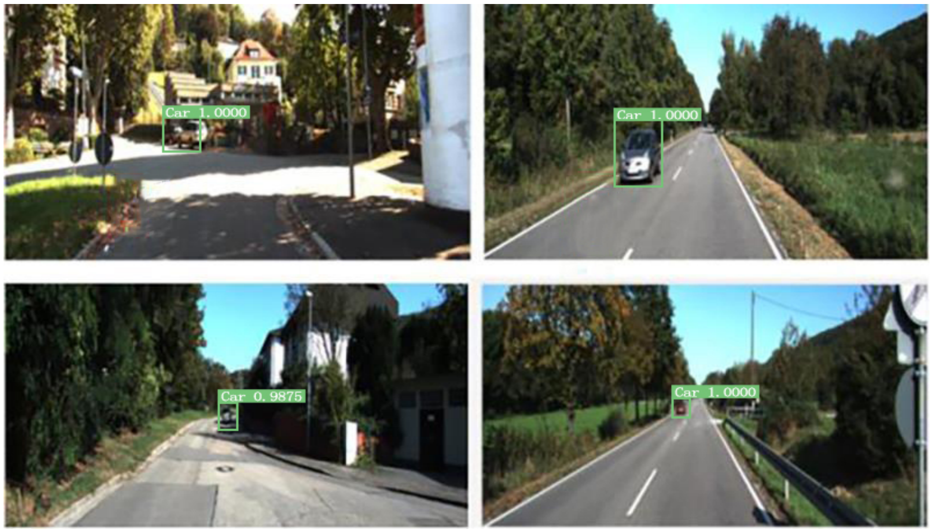

Figure 7: Example images of vehicle detection after adaptation in the first scenario

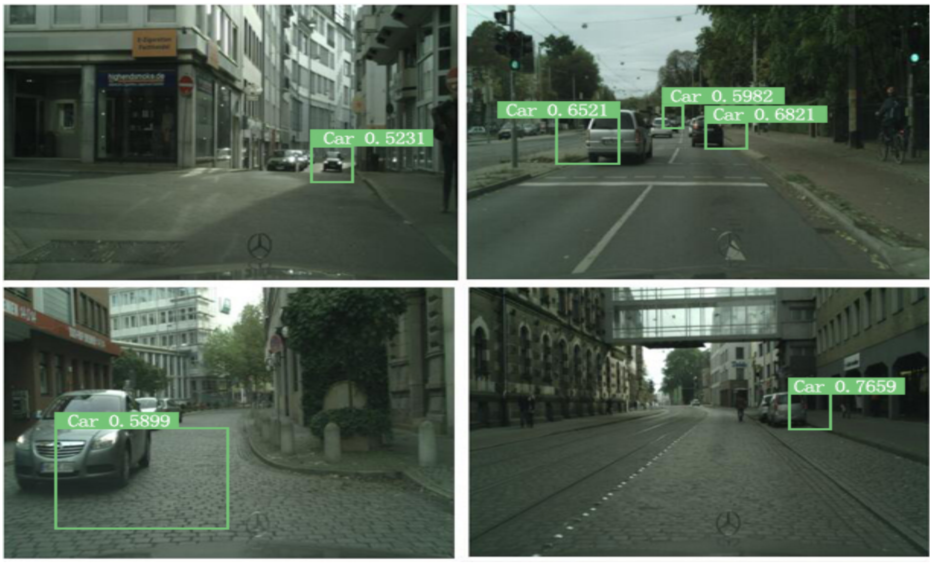

Figure 8: Example images of vehicle detection before adaptation in the second scenario

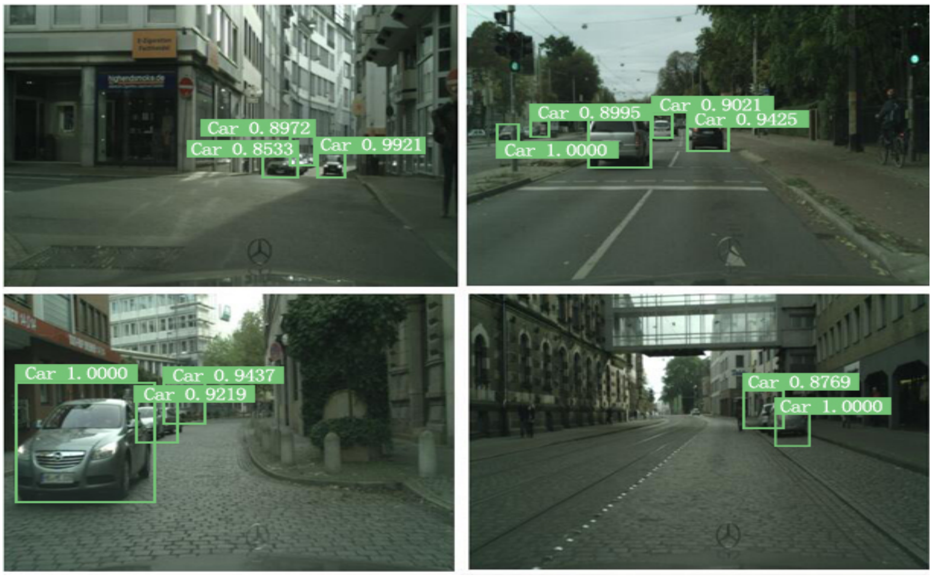

Figure 9: Example images of vehicle detection after adaptation in the second scenario

Figs. 10 and 11 show the comparison of the *PR* curves of different methods on two different datasets. It can be seen from the *PR* curves in Figs. 10 and 11 that the performance of the proposed method was superior to FRCNN and FRCNN in the wild [Chen, Li, Sakaridis et al. (2018)] in two different scenarios.

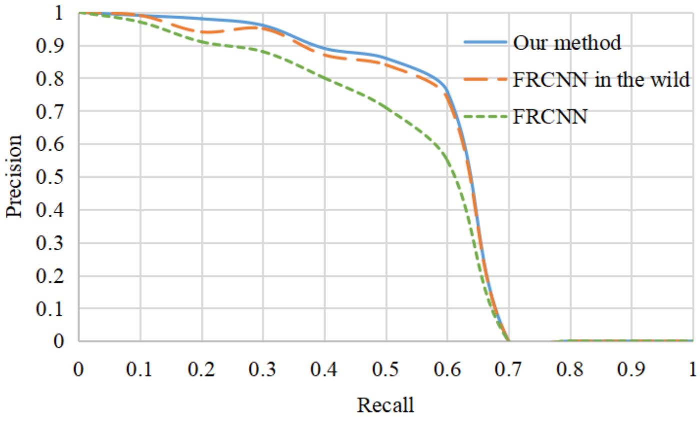

Figure 10: P-R curve in the first scenario

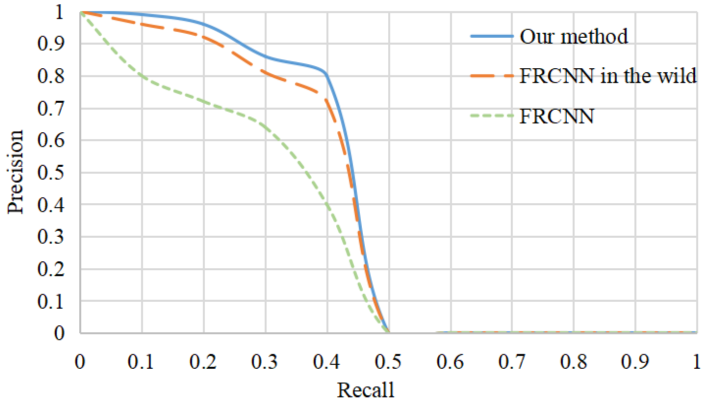

Figure 11: P-R curve in the second scenario

Four kinds of comparative experiments were conducted: 1) Only Faster RCNN model was used; 2) Faster RCNN model was used in combination with image space; 3) Faster RCNN model was used in combination with feature space; 4) Faster RCNN model was used in combination with image space and feature space (our method). The comparative experimental results in two different scenarios are shown in Tabs. 1-4.

Tabs. 1 and 2 show the comparison of experimental results of different methods in the first scenario, and Tabs. 3 and 4 are the result of the second scenario.

Table 1: Comparison of difference methods in the first scenario

Method	*AP* (%)
Faster RCNN	52.5
Faster RCNN+Image space	57.7
Faster RCNN+Feature space	58.5
Faster RCNN+Image space+Feature space	64.8

Table 2: Compared with other methods in the first scenario

Method	*AP* (%)
Faster RCNN	52.5
YOLOv3[Redmon]	51.8
FRCNN in the wild [Chen]	62.9
Our method	64.8

Table 3: Comparison of difference methods in second scenario

Method	*AP* (%)
Faster RCNN	28.8
Faster RCNN + Image space	36.6
Faster RCNN+ Feature space	37.4
Faster RCNN + Image space + Feature space	40.8

Table 4: Compared with other methods in the second scenario

Method	*AP* (%)
Faster RCNN	28.8
YOLOv3 [Redmon]	29.1
FRCNN in the wild [Chen]	38.5
Our method	40.8

According to the information in Tabs. 1-4, we can summarize the following conclusions:

(1) The improved model based Faster RCNN combined with image style transfer technology and domain adaptation theory improved the accuracy of vehicle detection in different scenarios. It proves that this method is effective in solving the problem of poor performance of the detection model due to the domain shift between the source domain and the target domain.

(2) The image style transfer technology and domain adaptive concept improved accuracy. In the first scenario, the image style transfer technique was used to improve the original model by 5.2%, the domain adaptation components based on the two feature maps improved the model by 6%, and both methods applied simultaneously improved the model by 12.3%. In the second scenario, the image style transfer technique was used to improve the contrast of the original model by 7.8%, the domain adaptation components based on the two feature maps improved the model by 8.6%, and both methods used simultaneously improved the model by 12%.

(3) Compared with the current state-of-the-art methods, our technique achieved the best result in different scenarios. The results of the experiments provide proof that our method is the most competitive in solving domain adaptive problems.

5 Conclusion

This work aimed to solve the problem of performance drop in the detection model due to the domain shift between the source domain and target domain in vehicle detection tasks. A new domain adaptive vehicle detection model based on feature space and image space was proposed that achieved good results in unsupervised vehicle detection domain adaptive tasks. The problem of domain adaptive vehicle detection in more complex scenarios will be studied in the future.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Bousmalis, K.; Silberman, N.; Dohan, D.; Erhan, D.; Krishnan, D. (2017): Unsupervised pixel-level domain adaptation with generative adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, pp. 7.

Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; Van Gool, L. (2018): Domain adaptive faster R-CNN for object detection in the wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3339-3348.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M. et al. (2016): The cityscapes dataset for semantic urban scene understanding. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213-3223.

Dumoulin, V.; Shlens, J.; Kudlur, M. (2016): A learned representation for artistic style. *International Conference on Learning Representations.* arXiv:1610.07629.

Everingham, M.; Eslami, S. M. A.; Van Gool, L.; Williams, C. K. I.; Winn, J. et al. (2015): The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98-136.

Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A. C. (2017). DSSD: Deconvolutional single shot detector. *Processing Systems*, pp. 137-144.

Ganin, Y.; Lempitsky, V. (2015): Unsupervised domain adaptation by back propagation. *International Conference on Machine Learning*, pp. 1180-1189.

Gatys, L. A.; Bethge, M.; Hertzmann, A.; Shechtman, E. (2016): Preserving color in neural artistic style transfer. arXiv preprint arXiv:1606.05897.

Gatys, L. A.; Ecker, A. S.; Bethge, M. (2016): Image style transfer using convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414-2423.

Geiger A.; Lenz P.; Stiller C.; Urtasun, R. (2013): Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231-1241.

Girshick, R. (2015): Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440-1448.

Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. (2014): Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J. Y.; Isola, P. et al. (2017): CyCADA: cycle-consistent adversarial domain adaptation. arXiv:1711.03213.

Johnson, J.; Alahi, A.; Li, F. F. (2016): Perceptual losses for real-time style transfer and super-resolution. *European Conference on Computer Vision*, pp. 694-711.

Li, P.; Liang, X.; Jia, D.; Xing, E. P. (2018): Semantic-aware grad-gan for virtual-to-real urban scene adaption. arXiv preprint arXiv :1801.01726.

Liu, M. Y.; Tuzel, O. (2016): Coupled generative adversarial networks. *Advances in Neural Information Processing Systems*, pp. 469-477.

Liu, W.; Anguelov, D.; Erhan D.; Szegedy, C.; Reed, S. et al. (2016): SSD: single shot multibox detector. *European Conference on Computer Vision*, pp. 21-37.

Long, M.; Cao, Y.; Wang, J.; Jordan, M. I. (2015): Learning transferable features with deep adaptation networks. arXiv:1502.02791.

Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P. S. (2013): Transfer feature learning with joint distribution adaptation. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2200-2207.

Long, M.; Zhu, H.; Wang, J.; Jordan, M. I. (2016): Unsupervised domain adaptation with residual transfer networks. *Advances in Neural Information Processing Systems*, pp. 136-144.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z. et al. (2017): Least squares generative adversarial networks. *IEEE International Conference on Computer Vision*, pp. 2813-2821.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; Yang, Q. (2011): Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199-210.

Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. (2016): You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788.

Redmon, J.; Farhadi, A. (2018): Yolov3: An incremental improvement. arXiv: 1804.02767.

Song, H. S.; Liang, H. X.; Li, H. Y.; Dai, Z.; Yun, X. (2019): Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*, vol. 11, no. 4, pp. 713-726.

Sun, B.; Feng, J.; Saenko, K. (2016): Return of frustratingly easy domain adaptation. *AAAI Conference on Artficial Intelligence*, vol. 6, no. 7, pp. 8.

Wu, H.; Liu Q.; Liu X. D. (2019): A review on deep learning approaches to image classification and object segmentation, *Computers, Materials & Continua*, vol. 60, no. 2, pp. 575-597.

Zhang, H.; Dana, K. (2017): Multi-style generative network for real-time transfer. arXiv:1703.06953.

Zhang, L. L.; Liu, J.; Luo, N. N.; Chang, X. J.; Zheng, Q. H. (2018): Deep semi-supervised zero-shot learning with maximum mean discrepancy. *Neural Computation*, vol. 30, no. 6, pp. 1647-1672.

Zhang, X. X.; Zhu, X. (2020): Moving vehicle detection in aerial infrared image sequences via fast image registration and improved YOLOv3 network. *International Journal of Remote Sensing*, vol. 41, no. 11, pp. 4312-4335.

Zhu, J. Y.; Park, T.; Isola, P.; Efros, A. A. (2017): Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv: 1703.10593.