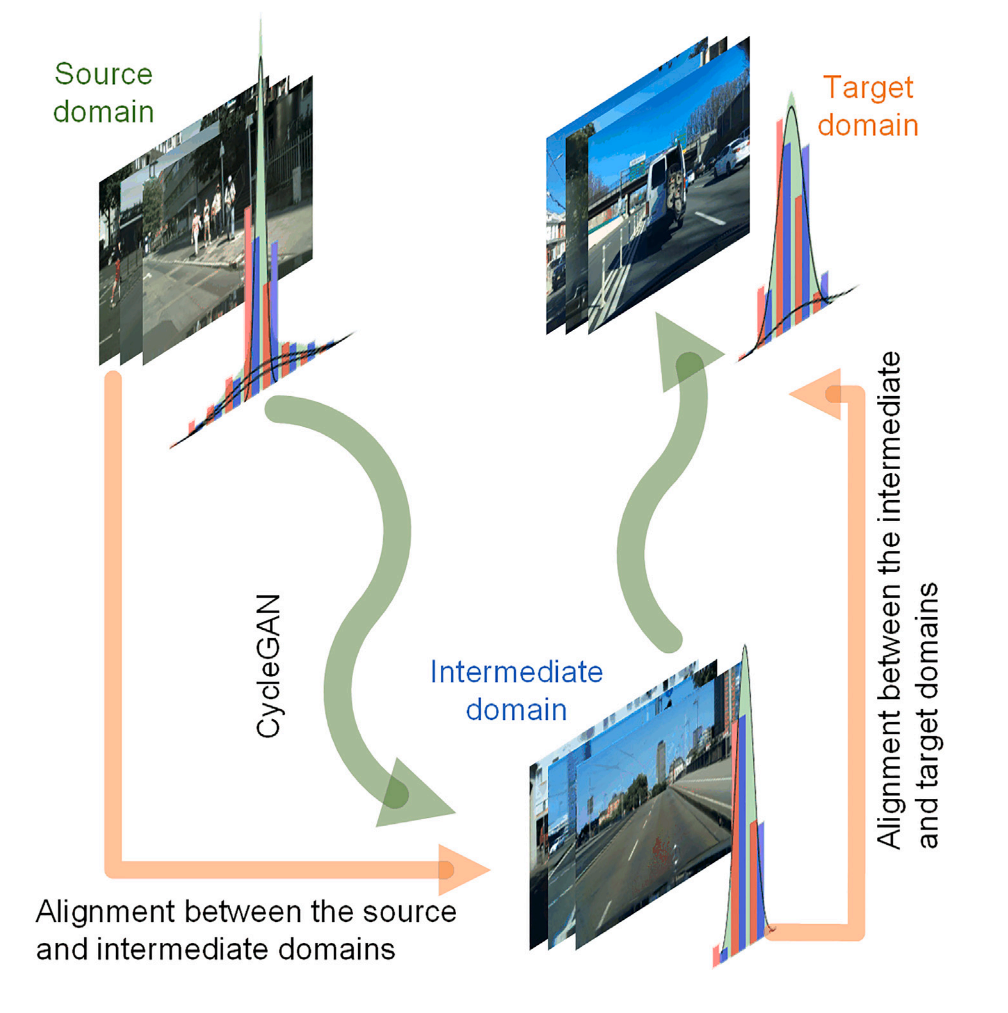
iScience

CellPress
OPEN ACCESS

Article

Cross-domain pedestrian detection via feature alignment and image quality assessment


Source domain
Target domain
CycleGAN
Intermediate domain
Alignment between the source and intermediate domains
Alignment between the intermediate and target domains

Jun Yao, Zhilin Guo, JunJie Yu, Nan Yan, Qiong Wang, Wei Yu

40516473@qq.com (J.Y.)
yuweits@cumt.edu.cn (W.Y.)

Highlights

Decompose a difficult large gap cross-domain task into two easier subtasks

Use an intermediate domain synthesized by an image-to-image translation network

Use image quality assessment to improve performance

Implemented cross-domain detection in multiple scenarios

Yao et al., iScience *27*, 109639
April 19, 2024 © 2024 The Author(s). Published by Elsevier Inc.
https://doi.org/10.1016/j.isci.2024.109639

iScience

CellPress
OPEN ACCESS

Article

Cross-domain pedestrian detection via feature alignment and image quality assessment

Jun Yao,[1,3,*] Zhilin Guo,[1] JunJie Yu,[1] Nan Yan,[1] Qiong Wang,[1] and Wei Yu[1,2,*]

SUMMARY

Datasets collected under different sensors, viewpoints, or weather conditions cause different domains. Models trained on domain A applied to tasks of domain B result in low performance. To overcome the domain shift, we propose an unsupervised pedestrian detection method that utilizes CycleGAN to establish an intermediate domain and transform a large gap domain-shift problem into two feature alignment subtasks with small gaps. The intermediate domain trained with labels from domain A, after two rounds of feature alignment using adversarial learning, can facilitate effective detection in domain B. To further enhance the training quality of intermediate domain models, Image Quality Assessment (IQA) is incorporated. The experimental results evaluated on Citypersons, KITTI, and BDD100K show that MR of 24.58%, 33.66%, 28.27%, and 28.25% were achieved in four cross-domain scenarios. Compared with typical pedestrian detection models, our proposed method can better overcome the domain-shift problem and achieve competitive results.

INTRODUCTION

With the development of deep learning and computer vision, the performance of object detection tasks has been greatly improved.[1,2,3–6] Meanwhile, a particular object detection task, pedestrian detection, plays an increasingly important role in autonomous driving, the Internet of Things, and security checks. The performance and robustness of traditional pedestrian detectors heavily rely on labeled training data, often collected in controlled environments with specific characteristics. However, deploying these detectors in real-world scenarios poses significant challenges due to the domain shift between the source and target domains. For instance, domains can differ in lighting, weather, camera viewpoints, and object scales, which further complicate the accurate detection of pedestrians.[7,8]

UDA (Unsupervised Domain Adaptation) has emerged as a promising approach to address the domain-shift problem in pedestrian detection. Unlike traditional approaches that require manually annotated data from the target domain, UDA methods aim to learn domain-invariant representations by leveraging the information from both the source and target domains. By aligning the feature distributions across domains, UDA methods facilitate better generalization and adaptability of pedestrian detectors in real-world scenarios. Numerous methods for image classification have been developed.[9–16] In contrast, the methods for semantic segmentation[17,18] and object detection[19,20,21] still need to be revised because higher annotation levels result in a significant gap between domains.

In this article, we aim to align the distribution between the source domain and the target domain,[11] we created an intermediate domain between the source and target through a generative adversarial network,[22] thereby avoiding direct mapping between two distributions with significant gaps. Specifically, we decompose the alignment problem into two small problems. Firstly, an image-to-image translation network transforms the source images into an intermediate domain with a similar appearance (lighting, weather, and so forth) as the target images.[22] We then align the source and intermediate distributions to construct an intermediate feature space, which is easier than aligning to the targets. Once this intermediate domain is aligned, we use it as a bridge to further connect to the target domain. Therefore, the proposed progressive adaptation through the intermediate domain decomposes the original alignment between source and target domains into two subtasks, solving an easier problem with smaller domain gaps.

Due to the use of unsupervised image-to-image translation networks for domain alignment, the translation quality of intermediate domain images is not consistent. To reduce the outlier impact of the low-quality images, an image quality assessment method is introduced to filter out the intermediate domain images that are too far from the target domain images.

We evaluated our method in various scenarios using KITTI,[23] Citypersons,[24] and BDD100k.[25] With the proposed domain adaptive pedestrian detection method, our method effectively improves performance in unsupervised domains obtaining the MR of 24.58%, 33.66%, 28.27%, and 28.25% in camera viewpoints and 3 types of weather domain adaptation tasks, respectively.

The main contributions of our work are summarized as follows: 1) we introduced a new framework using adversarial learning to achieve progressive feature alignment for pedestrian detection; 2) we used a progressive domain adaptation scheme, which involves aligning the

[1]The Engineering&Technical College of Chengdu University of Technology, Xiaoba Road, Leshan 614000, China
[2]China University of Mining and Technology, Daxue Road, Xuzhou 221116, China
[3]Lead contact
*Correspondence: 40516473@qq.com (J.Y.), yuweits@cumt.edu.cn (W.Y.)
https://doi.org/10.1016/j.isci.2024.109639

source and intermediate domains in the first stage, and the intermediate and target domains in the second stage; and 3) we introduced IQA to reduce the negative impact of low-quality images on intermediate domain model training and feature alignment.

Related work

Pedestrian detection

Pedestrian detection has been extensively studied in the computer vision community. Traditional methods typically employ handcrafted features and classifiers to detect pedestrians. Dollar et al.[26] proposed a refined per-frame evaluation method that measures performance related to scale and occlusion in detail, achieving the best performance of 16 pre-trained detectors in six datasets. Walk et al.[27] proposed a Self-similarity feature on the color channel introduced, which continuously improves the detection performance of still images and video sequences on different datasets, and is combined with HOG, which is 20% higher than the most advanced at that time. Benenson et al.[28,29] used existing detectors with mainly decision forests over hand-crafted feature outputs and re-scored them with plus-bounding box regression.[30–33] However, these methods heavily rely on manually extracted features and often suffer from limited generalization to different domains due to the lack of adaptability.

After 2016, deep learning-based approaches achieved remarkable success in pedestrian detection by leveraging the power of CNN (Convolutional Neural Network) and VIT (Vision in Transformer). Girshick et al.[1,6,34] proposed a region proposal method and utilized the CNN network to accelerate the generation process. Liu et al.[2,3–5] proposed a single-stage detection scheme, using a fixed set of predefined anchor boxes as proposals to further reduce the computational requirements for proposal generation. Lin et al.[35] proposed a deformable DETR detector to design a mechanism to leverage the less occluded visible parts of pedestrians specifically for pedestrian detection. Although these methods learn discriminative features directly from the data, they lead to improved performance. However, they depend on a large amount of labeled training data and the domain-shift problem still poses a challenge to their deployment in real-world scenarios.

Domain adaptation

In recent years, domain adaptation technology has improved the generalization ability of pedestrian detection models by eliminating domain offsets between labeled data in the source domain and unlabeled data in the target domain. Ganin et al.[10] proposed an adversarial learning method that utilizes a confusion domain classifier to obtain the ability to extract common features in both the source and target domains. Bousmalis et al.[9,15,19] used adversarial learning to narrow the feature distribution between the source and target domains. In addition, Zhu et al.[22,36] utilized an unpaired image-to-image conversion method to align pixel features of images in the domain. Tsai et al.[18,37] adopted adversarial learning in structured output spaces to solve the problem of feature/pixel space alignment.

Hoffman et al.[20] fine-tuned the fully supervised classification model for object detection, solving the domain adaptation problem of object detection in a weakly supervised manner. Inoue et al.[21] improved performance by fine-tuning the synthesized data using pseudo labels in the target domain. Chen et al.[19] proposed to narrow the domain gap at the image and instance levels through adversarial learning. Zhu et al.[38] excavated discriminative regions for comparison, strengthening the matching of local features to improve cross-domain detection performance. Saito et al.[39] focused on aligning the local receptive field of low-level features and the weak ratio of the global region. Kim et al.[40] utilized an image translation network to generate multiple domains and used a multi-domain discriminator to simultaneously adapt to all domains.

In the above works, we found that most of the intermediate feature alignment is achieved through simple adversarial learning or image-to-image translation. Excessive domain shift and low-quality image translation can affect the effectiveness of cross-domain alignment. To solve these problems, we introduce an intermediate domain to reduce the distance between domains and then use the image quality assessment method to filter out low-quality translations, thereby improving the performance of cross-domain pedestrian detection.

Methodology

We propose two subtasks to improve the cross-domain detection performance. The source, intermediate, and target domains are denoted as S, M, and T, respectively. The conventional cross-domain adaptation process from source domain S to target domain T is denoted as S → T, so our proposed adaptation subtasks are expressed as S → I and I → T. The main steps of our proposed adaptation framework are shown in Figure 1. We use three components to implement pedestrian detection while aligning the feature spaces of the source, intermediate, and target domains. They are the adversarial learning network denoted as A, the unsupervised image-to-image translation network denoted as G, and the IQA process denoted as I. All details will be discussed in subsequent sections.

Domain adaptation via adversarial learning

We use a deep learning framework to detect and align distributions in the feature space which consists of a feature extractor, a detector, and a discriminator. We adopt the Cascade R-CNN[41] composed of a sequence of detectors trained with increasing IoU thresholds for pedestrian detection tasks. Cascade R-CNN achieves state-of-the-art performance on the COCO dataset and significantly improves high-quality detection on generic and specific object detection datasets, including VOC, KITTI, Citypersons, and WiderFace. It has a base encoder E and a feature extractor F through where the image features denoted feature map E(I) were extracted and fed into two branches: Region Proposal

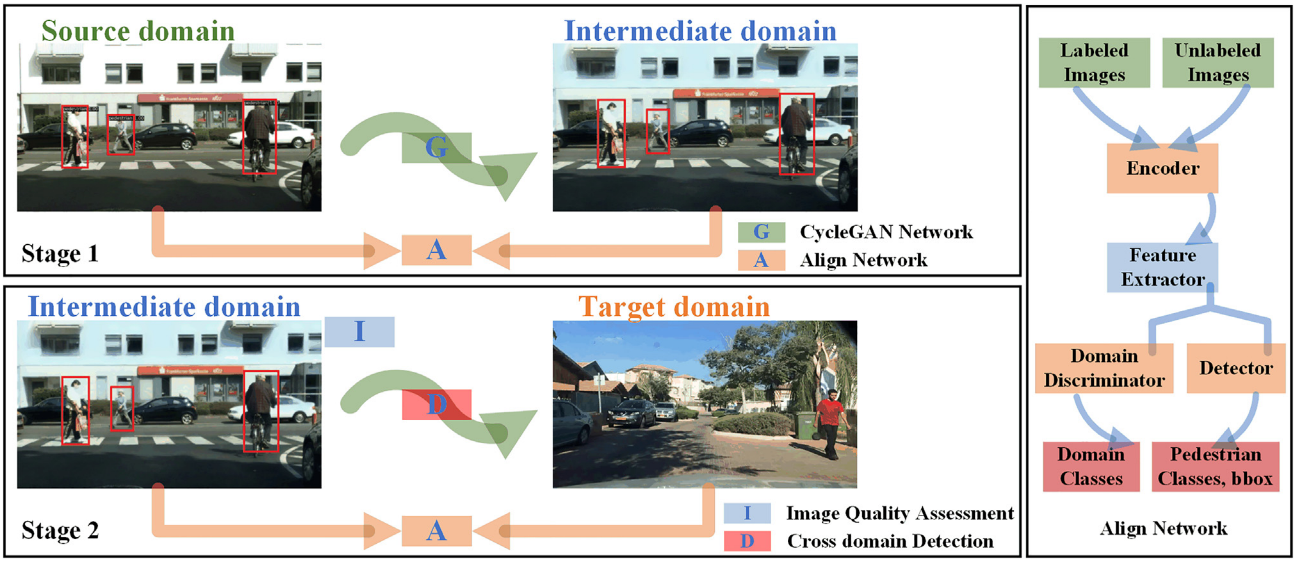


Figure 1. The main steps of our proposed domain adaptation framework

The framework consists of adaptation stage 1 and stage 2. In stage 1, we transform source images to intermediate ones by a CycleGAN network G. Afterward, we perform the first alignment to the intermediate domain via labeled source images. In stage 2, the framework applies a second alignment which takes the intermediate distribution with labels inherited from the source and aligns the features with the target distribution. In addition, an IQA process is applied before detector D to filter out the low-quality images transformed by G. All stages are both passed through the align network A to extract domain invariant features in an adversarial manner.

Network (RPN) and Region of Interest (ROI) classifier. As shown in Figure 1, the two branches output categories and detection boxes to be the detector. The loss function of the detector L_{det} is defined as Equation 1:

$$L_{det} = L_{rpn} + L_{cls} + L_{reg} \qquad \text{(Equation 1)}$$

where L_{rpn}, L_{cls}, and L_{reg} are the loss of the RPN, classifier, and bbox regression, respectively. All further details refer to the original article.[41]

To align the distributions, we append a domain discriminator D after the feature extractor F. This branch is to discriminate which domain the feature E(I) is from. This discriminator gives us the probability of each image belonging to the target domain P = D(E(I)). Then a binary cross entropy loss based on the domain label d is applied to P. The discriminator loss function L_{dis} is defined as Equation 2:

$$L_{dis} = -\sum dlogp + (1\text{-}d)log(1\text{-}p) \qquad \text{(Equation 2)}$$

The main method of adversarial learning is to use gradient reverse layers (GRL)[42] to learn domain invariant features E(I). GRL performs positive gradient updates for the detector and negative gradient updates for the discriminator. As a result, the feature extractor F receives gradients that force it to update in an opposite direction which maximizes the discriminator loss, thereby confusing the discriminator to distinguish which domain the image comes from. For the domain adaptation task S → T, given source images I_S and target images I_T, the overall loss L_{all} is defined as Equation 3:

$$L_{all} = L_{\text{det}}(E(I_s)) + \lambda(L_{dis}(E(I_s)) + L_{dis}(E(I_T))) \qquad \text{(Equation 3)}$$

where λ is a weight applied to balance the loss of the discriminator. Specifically, because there are align networks in both stages, the I_S and I_T represent the source and intermediate domain images in stage 1, and the intermediate and target domain images in stage 2.

Progressive adaptation via image-to-image translation network

Directly aligning feature distributions between two distant domains often results in poor performance. So, we create an intermediate domain that plays a bridging role in achieving progressive domain adaptation.

CycleGAN[22] is an excellent generative adversarial network for creating an intermediate distribution. It synthesizes target distribution at the pixel level and can achieve bidirectional intermediate domain conversion, so it can achieve our innovation point: our proposed method also improves performance by swapping source and target domain. The fundamental motivation behind this is that the intermediate domain I created by CycleGAN has the same content and different appearance styles as source domain S, while intermediate domain I and target domain T have the same pixel-level feature distribution but different content. Therefore, the challenge of a large domain gap between S and T can be effectively reduced by an intermediate domain with a bridge role. Figure 2 shows feature space distribution visualization examples using the BDD100K, KITTI, and Cityspensons datasets. We use a dimensionality reduction algorithm t-SNE[43] to plot the feature map E(I)

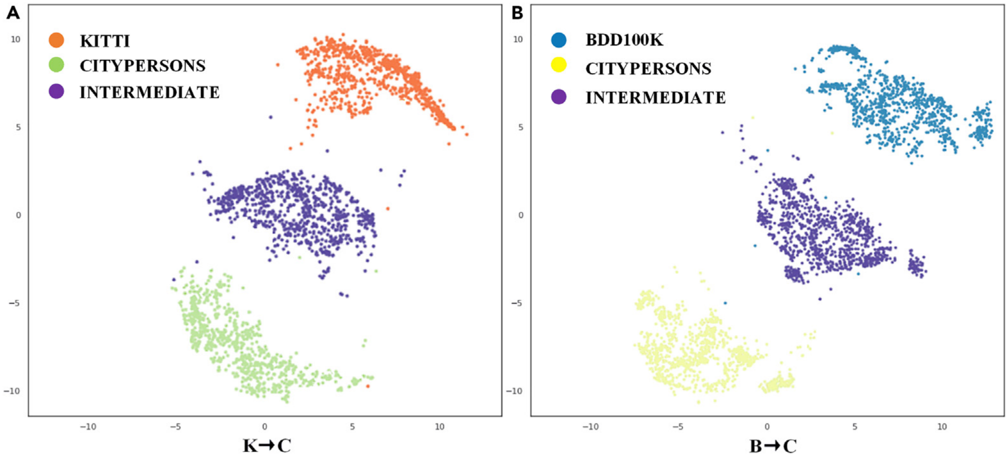


Figure 2. t-SNE visualized the intermediate feature space distribution between source and target domains

In (A) shows the intermediate distribution (purple) from KITTI (orange) to Citypersons (green).

In (B) shows the intermediate distribution (purple) from BDD100K (blue) to Citypersons (yellow).

as a 2-dimensional distribution. As shown, the distribution of the intermediate domain is exactly lying in the distributions of source and target domains.

When training the CycleGAN model, we need to specify a root directory and then store the source domain images in the folder train_A and test_A. Store the target domain images in the folder train_B and test_B. The source and target can be swapped because a CycleGAN model can generate images of the style of each domain participating in training. Then we set Buffer_ Size = 50, and randomly sample 50 images to participate in each update. The loss function consists of 2 cycle_loss and 2 id_loss with weights of 10 and 0.5. We use the Adam optimizer, where the moment estimates for both stages are set to β1 = 0.5, β2 = 0.999. The image normalization standard is mean = [0.5,0.5,0.5], std = [0.5,0,0.5]. The evaluation metrics for the generator are FID and IS. The total number of training Epochs is 80000. Because image translation does not require too much data augmentation, we only used Crop, Resize, and Horizontal Flip. In the generation phase, we need to turn off Crop and Horizontal Flip and specify the target domain name. We have summarized all the details in Table 1.

Therefore, we decompose a difficult large gap domain adaptation problem into two small gap domain alignment problems. The alignment method involves adversarial learning with an align network between two small gap domains.

Performance supervision via image quality assessment

We found that some dots of the intermediate domain are far away from both the source and target domains in Figure 2. The inconsistent quality of synthesized images causes these phenomena. In Figure 3, some images fail to preserve original details or contain extra shadows during translation, and these failure cases have a larger distance to target images. The first row shows examples from BDD100K synthesized to Citypersons, while the second and third rows show images from Citypersons to BDD100K.

Table 1. CycleGAN training statistics

Parameters		Details
Augmentation	Crop	256
	Resize	1024/512
	Horizontal Flip	*p* = 0.5
CPU/GPU/RAM		I7-6700/NVIDIA GTX 2080TI/16G
Environment		Python3.6.4/Pytorch1.10/OpenCV4.5.4
Buffer_size		50
Loss function		cycle_loss(weight = 10)+id_loss(weight = 0.5)
Optimizer		Adam(β1 = 0.5, β2 = 0.999)
Normalize		mean = [0.5, 0.5, 0.5]/std = [0.5, 0.5, 0.5]
Metrics		FID/IS
Epochs number		80000

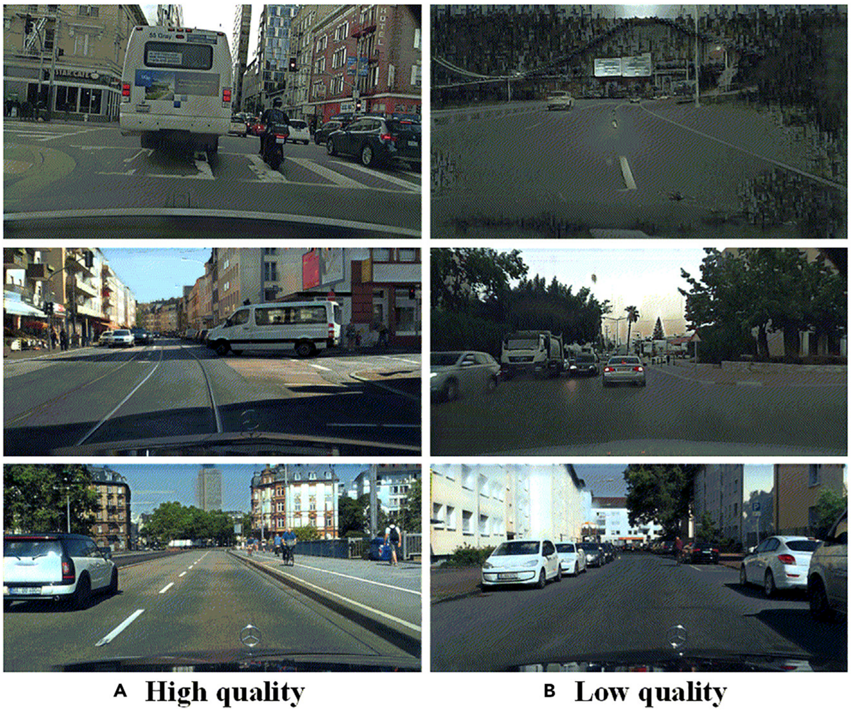


Figure 3. The image quality examples from the BDD100K and Citypersons datasets were translated to styles of each other

(A) Better quality translations.

(B) The image is in extra shadows that fail to preserve the original details of the cars and buildings, almost integrated with the background.

As the verification in section 4, when using these low-quality images to train the detection model in intermediate domain I and perform detection task I → T, these low-quality images can lead to cross-domain incorrect feature alignment. To alleviate this problem, we propose a filtering strategy based on the distance between intermediate domain images and target domain images. Specifically, the outliers that reach a certain distance from target images will be filtered out of the training set of the intermediate domain. We determine distance using the output score predicted by the domain discriminator D in the align network. This discriminator is trained to get the probability of each image belonging to the target domain, in which the predict function of the discriminator is defined as Equation 4:

$$D_{pred} = \frac{P_t(E(I))}{P_s(E(I))+P_t(E(I))} \qquad \text{(Equation 4)}$$

where E(I) is a feature map obtained from the intermediate domain image through the feature extractor F, and $P_t(E(I))$ and $P_s(E(I))$ are the probabilities that E(I) belongs to the source domain and target domain, respectively. Here, the higher the D_{pred} score achieved, the closer the distance to the target domain. On the other hand, low-quality images that are too far away from target domain images will be considered outliers and filtered out.

RESULTS

In this section, we evaluate performance in real-world scenarios through two typical cross-domain problems: 1) camera viewpoints adaptation and 2) weather adaptation to validate our proposed method.

For each adaptation scenario, we compare with a Faster R-CNN baseline model (trained on source domain dataset), a fully supervised model (trained on target domain dataset) denoted Oracle, and a model trained on the intermediate domain using our domain adaptation method, to show the effectiveness of unsupervised domain adaptation for cross-domain pedestrian detection.

Implementation details

In the experiments, we adopt Cascade R-CNN[41] for the detection network. The discriminator D uses 4 convolution layers with 3 × 3 filters. There are 64 channels in the first three layers, each followed by a ReLU function.[44] The last layer has 1 channel to output the prediction for domain discrimination. The intermediate domain is generated by CycleGAN[22] training on both source and target domain images.

Table 2. The partition of the dataset used in the experiments

Merge	A: Citypersons		B: BDD100K		C: KITTI	
Pedestrian	Pedestrian		Person		Pedestrian	
	Rider		Rider		Cyclist	
	Sitting Person		–		Person Sitting	
	Other Person		–		–	
	People Group		–		–	
	Ignore Region		–		–	
	Train	Val	Train	Val	Train	–
	2975	500	70000	10000	7481	–
Total	3475		80000		7481	
Persons	19654		86047		6336	
Per image	7.0		1.2		0.8	

To balance the discriminator loss with the detection loss in stage 1 and stage 2, we set the weight λ to 0.03 and 0.1, respectively. Our proposed method is implemented on the PC with one GTX2080Ti GPU, 16 GB memory, and one I7-6700 CPU using the PyTorch framework running on the Ubuntu18.04 Operate System.

Datasets

Citypersons. The Citypersons dataset[24] is a subset of the Cityscapes dataset[45] that only contains annotations about persons. There are 2975 images for training and 500 images for validation. The average number of pedestrians in an image is 7, providing a visible area and full body annotation. To unify the annotations of the detection objects, we merged all the annotations (ignore regions, pedestrians, riders, sitting persons, other persons with unusual postures, and groups of people) as Pedestrian.

BDD100k. The BDD100k dataset[25] from the University of California, Berkeley consists of over 100K video sequences, containing image level markers, object bounding boxes, drivable areas, lane markers, and full frame instance segmentation. There are 70000 images for training and 10000 images for validation. The dataset has geographical, environmental, and weather diversity, which is useful for training models and can therefore reduce the impact of environmental factors on identification results. The annotations of car, truck, bus, train, motorcycle, bicycle, traffic light, and traffic sign are discarded, while pedestrians and riders are retained as Pedestrian.

KITTI. A total of 7,481 images were collected while driving a data acquisition vehicle in highways, cities, and rural areas in the KITTI dataset.[23] All the images belong to the training set, so we only use KITTI as a source domain in the following experiments. Also, we discarded the annotations of Car, Van, Tram, Truck, Misc, and DontCare, and unified Pedestrian, Person Sitting, and Cyclist as Pedestrian.

In summary, we present the details of all the datasets used in Table 2. To achieve a unified detection task, we only retained annotations related to pedestrians in each dataset.

Adaptation for camera viewpoints

The underlying camera viewpoints and mechanisms lead to a critical domain shift in pedestrian detection tasks. In this section, we attempt to solve the first cross-domain detection problem using our proposed method due to differences in style and content caused by different camera viewpoints.

Firstly, the KITTI training set is used to train a baseline Faster R-CNN model and evaluated on the Citypersons validation set. Then, we apply an intermediate domain generated by CycleGAN for Faster RCNN to evaluate. Finally, we use our proposed method (Ours) to validate the effectiveness of the intermediate domain and IQA processes using an ablation study, respectively.

Unlike conventional object detection, the evaluation metrics for pedestrian detection use MR[24,26,28,33,35,46–49] instead of mAP. Miss rate is a term used to evaluate the performance of pedestrian detection algorithms. It refers to the rate at which the algorithm fails to detect a pedestrian correctly per image. The lower the miss rate, the better the performance of the algorithm. The MR can be represented as Equation 5:

$$MR = 1 - \frac{TP}{TP+FN} \qquad \text{(Equation 5)}$$

where *TP* (True Positive) predicts a positive sample and the result is correct, and *FN* (False Negative) predicts a negative sample, but the result is incorrect. We can determine the ratio of missed detection boxes in all images using this equation, as shown in Table 3.

The results showed that using a model trained on the source domain to solve the pedestrian detection task in the target domain achieved a very poor performance (Faster R-CNN). When using an intermediate domain to train models created by CycleGAN, the MR was improved by

Table 3. Camera viewpoints Cross-Domain Adaptation

Method	Miss Rate KITTI → Citypersons
Faster R-CNN	47.20
Faster R-CNN(W/CycleGAN)	38.12
Ours(W/CycleGAN)	25.16
Ours(W/CycleGAN & IQA)	**24.58**
Oracle	12.37

9.08% (Faster R-CNN with CycleGAN). Furthermore, using our proposed method to train the model in the intermediate domain reduces the MR by 12.96%. Finally, using the IQA process, the MR was reduced to 24.58%, which is closer to the fully supervised model directly trained on the target domain, with an MR of 12.37%.

Furthermore, FPPI (False Positive Per Image) was used to analyze the MR information under different false positive in each image. FPPI is commonly used in pedestrian detection performance evaluation and displays the MR situation under different FPPI values by drawing an FPPI-MR curve. Usually, we hope to achieve lower MR values under lower FPPI values, which means the model can accurately detect targets while minimizing false detections as much as possible. Figure 4 shows the FPPI performance under different models on the cross-domain detection task KITTI → Citypersons.

Adaptation for weather

To apply pedestrian detection models to different weather conditions in real-world scenarios, this section proposes weather adaptation from multiple weather conditions. The Citypersons dataset[24] and BDD100K dataset[22] are used as source and target domains, respectively, and then exchange the source and target.

Table 4 shows that the model trained on the intermediate domain effectively reduces the domain gap with the target domain under different weather conditions through our proposed domain adaptation method. Due to the higher difficulty of the pedestrian detection task, we have demonstrated competitive performance compared to state-of-the-art cross-domain detection methods.[19,38,39,40] Compared with the baseline method, the Fater R-CNN using intermediate domains reduced MR by 6.15%, 5.06%, and 4.20%, respectively, from overcast, snowy, and rainy weather to clear weather of Citypersons. Our proposed domain adaptation method achieved further MR reductions of 3.54%, 7.84%, and 7.77%, respectively. At the same time, from the various weather conditions of BDD100K to the clear weather of Citypersons, the final MR reached 22.30%. We noticed that during the IQA process of task C → B: overcast using Ours, the performance decreased due to the significant difference in the number of images between source and target domains. After the IQA process, the difference in sample size was further expanded, resulting in a shortage of features extracted by the deep learning model. Therefore, the intermediate domain by image translation has been tightly distributed to the target domain and inherits annotations from the source domain for learning. In summary, this experiment proves adaptation to weather conditions and the distribution alignment process, resulting in our method being very close to Oracle results.

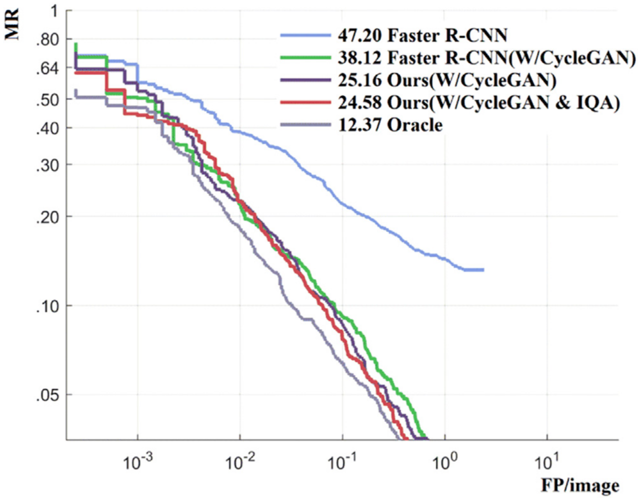


Figure 4. Comparison of FPPI results on the cross-domain detection task KITTI → Citypersons

Oracle and Ours dropped to a minimum value, while Faster R-CNN models further reduced MR as the FPPI value increased. Especially when the baseline model has a larger FPPI, MR is still not low enough.

Table 4. Weather cross domain adaptation

	Miss Rate			
Method	C → B: overcast	C → B: snowy	C → B: rainy	B → C
Number of images	2975 → 8770	2975 → 5549	2975 → 5070	70000 → 500
Faster R-CNN	43.35	41.17	40.22	28.87
Faster R-CNN(W/CycleGAN)	37.20	36.11	36.02	26.33
Ours(W/CycleGAN)	**32.05**	30.16	30.18	25.36
Ours(W/CycleGAN & IQA)	33.66	**28.27**	**28.25**	**22.30**
Oracle	18.86	17.52	17.36	12.37

Figure 5 shows the FPPI performance under 4 different weather cross-domain scenarios. We found that task BDD100K → Citypersons displayed the best FPPI-MR curve due to BDD100K having the biggest size, fully utilizing the common features of different pedestrian detection datasets, and aligning them to the smallest dataset for cross-domain detection. The other 3 show the FPPI performance of 3 weather adaptation tasks from Citypersons to BDD100K.

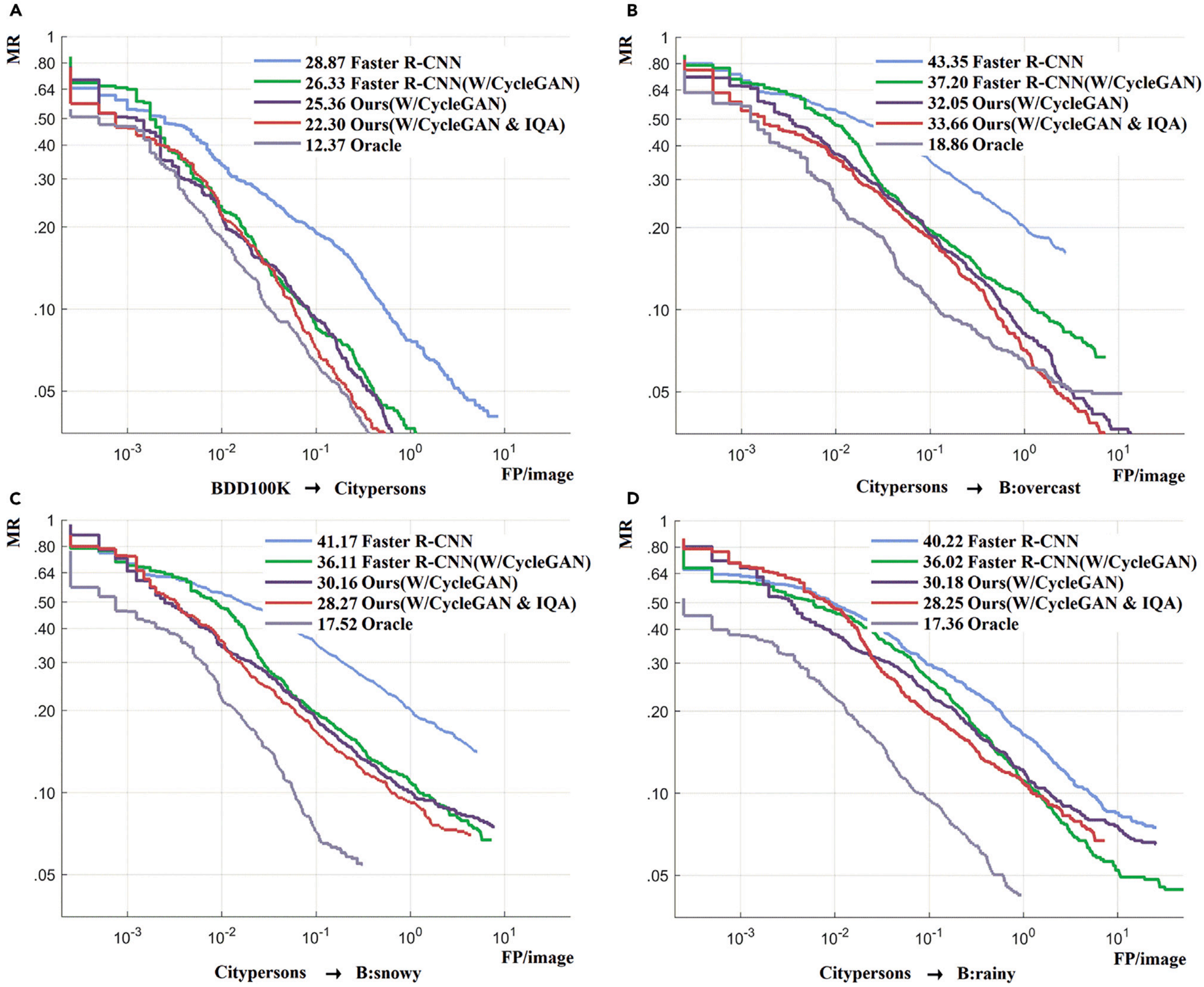


Figure 5. Comparison of FPPI results on four cross-domain detection tasks

Comparison of FPPI results on the cross-domain detection task (A) BDD100K → Citypersons, (B) Citypersons → B: overcast, (C) Citypersons → B: snowy, (D) Citypersons → B: rainy.

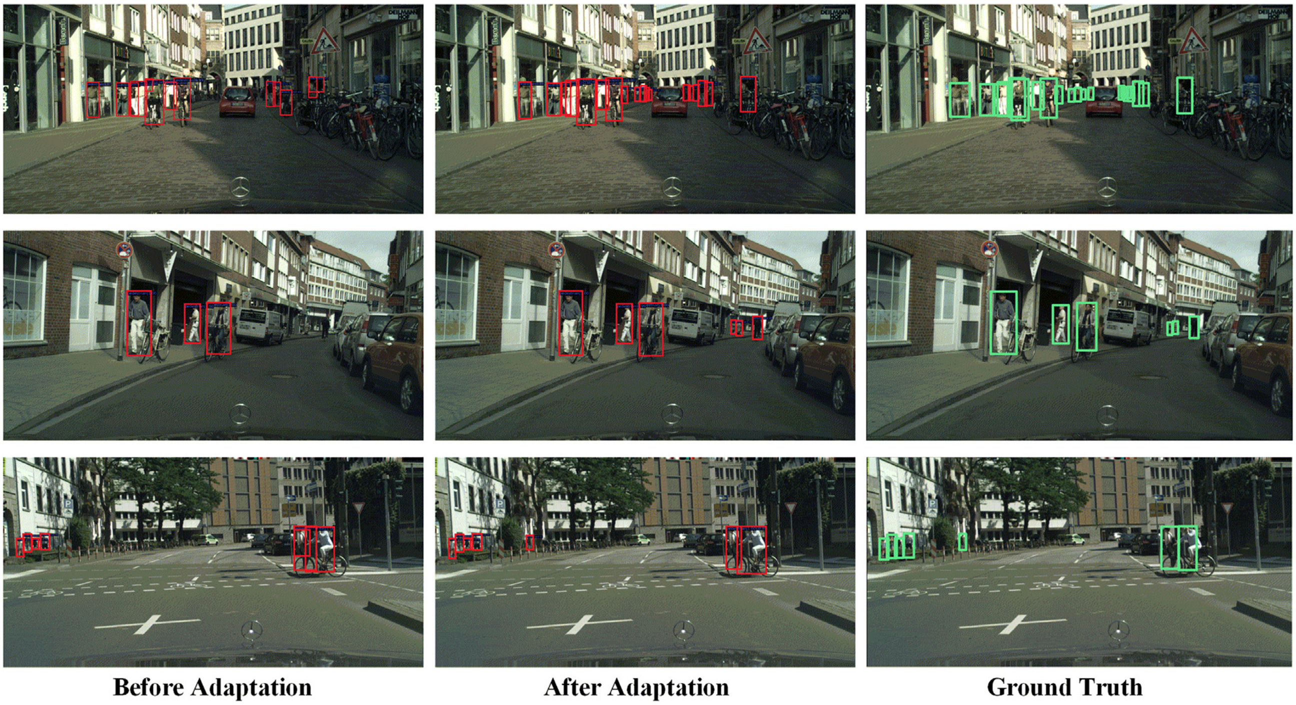


Figure 6. The results in cross-domain detection tasks K → C and B → C

The 3 columns represent detection results before adaptation, after adaptation, and ground truth of the target domain, respectively.

The similar size of images between the task Citypersons → B: snowy and Citypersons → B: rainy results in similar performance compared to task Citypersons → B: overcast.

More discussion

In Figure 6, The first row shows the detection results in the weather adaptation task B → C, and the last two rows show the detection results in the camera viewpoints adaptation task K → C.

We found that it is easy to overcome the domain-shift problem caused by viewpoints when crossing domains from big to small, and the detection results are consistent with ground truth. With the completion of the domain-adapted process, the previously undetected small targets and the overlapping population of false detection are correctly detected. The main reason is that the domain gap's impact on the detection targets can easily be compensated by more features learned from more images.

Figure 7 shows the detection results of 3 weather adaptation tasks from Citypersons to BDD100K. Adaptation to snowy and overcast is good, except for some small targets with wrong boxes regression. In rainy adaptation, there will be more false positives and missed boxes in a crowd, mainly due to the lens coating and the obstruction of rain.

As an extension of this study, we further tested the time adaptation scenarios from Citypersons to BDD100K to verify the effectiveness of our proposed unsupervised domain adaptive pedestrian detection method for domain gap issues caused by time changes. Figure 8 shows very similar results to the previous two figures. Our proposed domain adaptation method can apply detection models to unsupervised pedestrian detection tasks and obtain results very close to manual annotations.

Table 5 shows all results of time adaptation from Citypersons to BDD100K. We found that the scenes improvement at night and dawn/dusk are most noticeable, mainly because Citypersons is mostly collected in daytime environments, while night and dawn/dusk samples are very rare.

Comparison with state-of-the-art

In this section, we compare our work with state-of-the-art works on multiple datasets. To conduct a more comprehensive analysis, we continued to perform domain adaptation tasks on three pedestrian detection datasets: Caltech,[26] Foggy Cityscapes,[45] and WiderPedestrian.[50] Moreover, we use the AP (Average Precetion) to extract the pedestrian category from the best object detection works for comparison. Table 6 summarizes these comparison results.

To compare with other works using MR as an evaluation metric, such as CascadeRCNN,[46,49] SAN,[48] and HQATrans,[47] the results are shown in the left half of Table 6. Ref.[46,49] both used CascadeRCNN as the detection network and other additional public datasets as intermediate

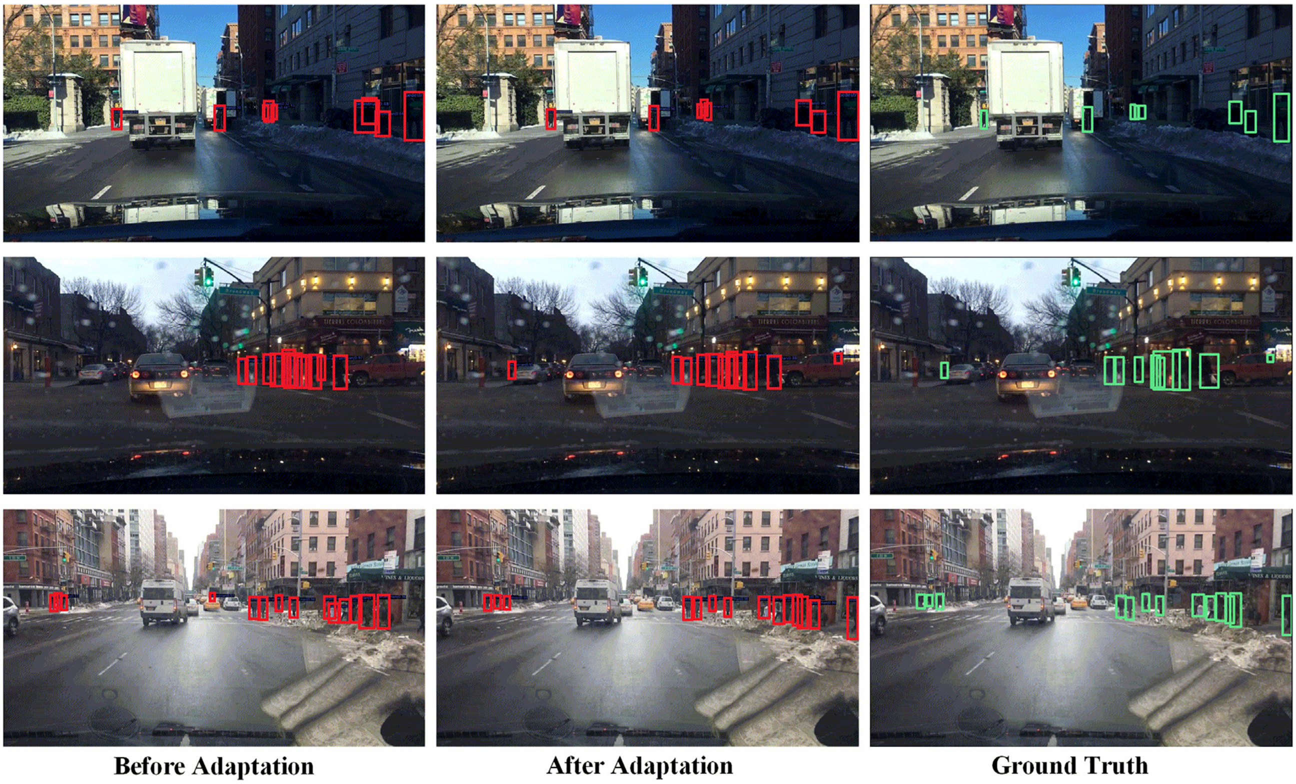


Figure 7. The results in cross-domain detection task C → B

The 3 rows represent the cross-domain adaptation detection results from clear days to snowy, rainy, and overcast days, respectively.

domains for progressive domain adaptation. Compared with the MR in ref. 51, Ours is 1.3 points ahead in task C → C and 1.95 points behind in task W → C without additional annotation datasets. Meanwhile, due to the advantages of frameworks and methods, we have achieved significant leading performance compared to ref. 52, 53, and 54 in two tasks.

AFAN,[51] SCD,[38] CycleConfusion,[52] and FasterRCNN[19] detect multiple categories, so we extract pedestrian categories from them and use AP as an evaluation metric for comparison. The results are shown in the right half of Table 6. Ref.[51] used an enhanced feature alignment network and an intermediate domain for domain adaptation, achieving an AP performance very similar to Ours in tasks C → F and C → K. Ref. 55 introduced an instance-level temporal cycle confusion and achieved the highest AP performance of 45.80% in task C → F, demonstrating the potential of self-supervised learning in domain adaptation tasks. Ref. 3 used two components to alleviate domain differences at the image and instance levels, achieving the best AP of 64.10% in Task C → K. The time adaptation tasks on BDD100K show our leading specific values ahead of CycleConfusion,[52] with a lead of 6.4 points in task D → N and 7.6 points in task N → D, respectively. Time variation introduces domain shift caused by the light condition, therefore, pixel-level progressive domain adaptation has more advantages than self-supervised image-level augmentation.

To sum up, our proposed method achieved competitive results in multiple cross-domain tasks using MR and AP as evaluation metrics. In the specific three tasks of C → C, D → N, and N → D, the best performance was achieved compared to other state-of-the-art works.

DISCUSSION

In this article, we propose a two-stage progressive unsupervised cross-domain pedestrian detection method, which uses the intermediate domain created by CycleGAN to bridge domain gaps and decompose a difficult large gap cross-domain task into two easier small cross-domain subtasks. Using the intermediate domain, our method first aligns the feature distribution from a source domain to an intermediate domain, and then from the intermediate domain to the target domain.

As we know, in driving safety, pedestrians are obstacles with the highest level of safety. Therefore, it is necessary to deploy a pedestrian detection system with domain adaptation capability in the autonomous driving system using one or more cameras. We need to carefully inspect the output of the original system and effectively integrate the input of pedestrian detection into the integrated perception system without interference.

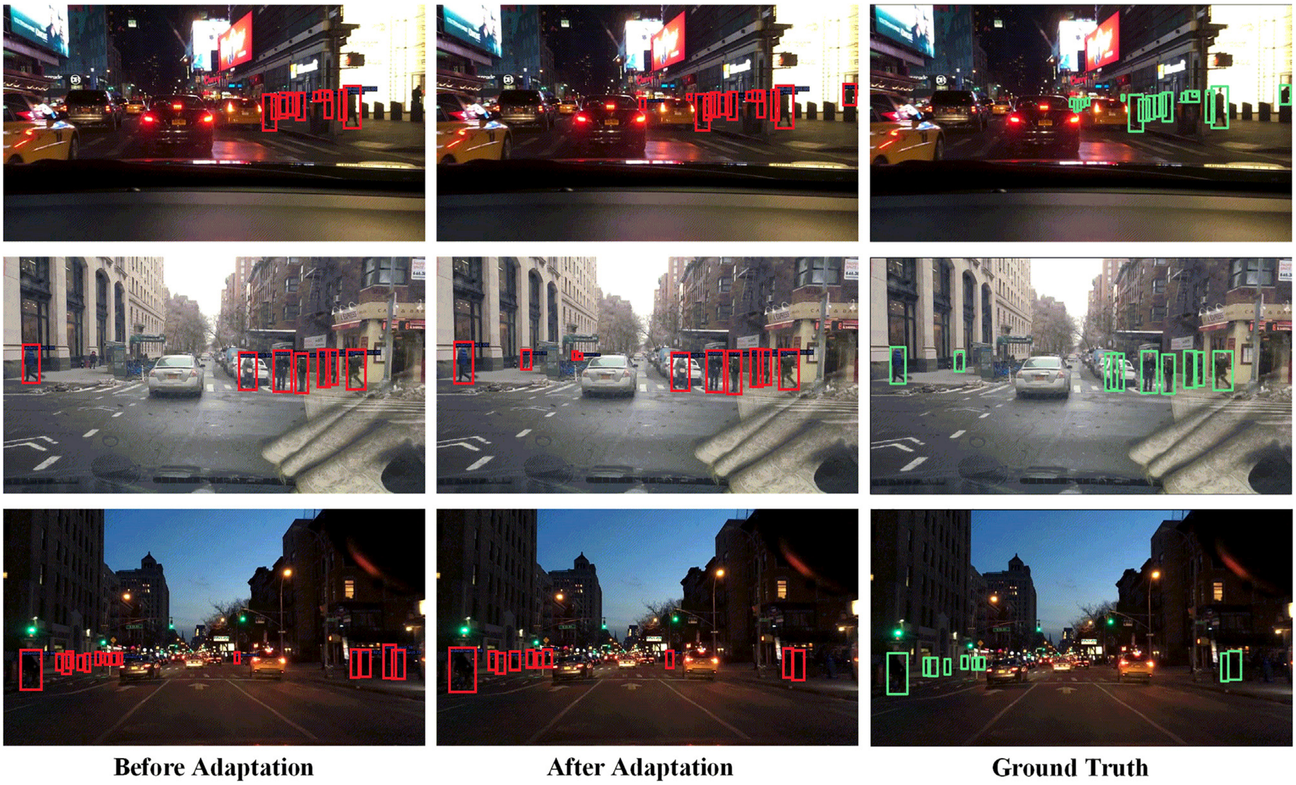


Figure 8. The results in time adaptation tasks from C → B
The 3 rows represent the cross-domain detection results from any time in Citypersons to night, daytime, and dawn/dusk in BDD100K, respectively.

Here are some road perception systems that can be combined with domain adaptive pedestrian detection: driving scene reconstruction, bird's-eye view object detection, bird's-eye view map segmentation, semantic occupancy prediction, multi-view depth estimation, and multi-modal object detection. Meanwhile, the following challenges will come through: 1) movement&acquisition, such as potential blurring caused by vehicle movement; 2) data processing issues that occur due to hardware failures, such as noise and quantization; 3) the problem of integrating 2D and 3D targets, such as how to model pedestrians and 3D occupied grids uniformly.

In extreme weather and lighting environments, we need more effective image preprocessing techniques to eliminate noise; Extreme changes in camera viewpoints can also lead to measurement failure, requiring more diverse and delicate anchor boxes to detect small, obscured, or tilted targets. Our proposed domain adaptation scheme can be extended to more cross-domain detection tasks, such as vehicles, crops, mineral products, residues, industrial defects, and so forth. However, in special fields such as medical imaging, and laser-plasma, the intermediate domain generated by CycleGAN cannot be used for domain adaptation. The size and shape of lesions require strict medical definitions, and plasma also has strict variation limits during discharge. CycleGAN is likely to produce image results that do not match the actual situation, making the detection results unreliable. Figure 9 shows some failed cases.

Table 5. Time cross domain adaptation

	Miss Rate			
Method	C → B: night	C → B: daytime	C → B: dawn/dusk	B → C
Number of images	2975 → 27971	2975 → 36728	2975 → 5027	70000 → 500
Faster R-CNN	47.43	42.55	48.12	28.87
Faster R-CNN(W/CycleGAN)	42.28	36.11	46.17	26.33
Ours(W/CycleGAN)	35.33	32.10	37.27	25.36
Ours(W/CycleGAN & IQA)	**31.56**	**30.52**	**32.71**	**22.30**
Oracle	18.28	16.32	18.12	12.37

Table 6. Comparison with state-of-the-art works

	Miss Rate (↓)		Average Precision (↑)			
Methods	W → C	C → C	C → F	C → K	D → N	N → D
AFAN[51]	–	–	42.50	57.70	–	–
CascadeRCNN[46]	**29.20**	36.50	–	–	–	–
SCD[38]	–	–	33.90	–	–	–
SAN[48]	–	44.17	–	–	–	–
CascadeRCNN[49]	39.70	–	–	–	–	–
CycleConfusion[52]	–	–	**45.80**	–	19.90	19.57
FasterRCNN[19]	–	–	25.00	**64.10**	–	–
HQATrans[47]	–	51.28	–	–	–	–
Ours	31.15	**35.20**	40.97	60.55	**26.30**	**27.17**

W → C: WiderPedestrian to Citypersons C → C: Caltech to Citypersons
C → F: Citypersons to Foggy-Cityscapes C → K: Citypersons to KITTI
D → N: Daytime to Night on BDD100K N → D: Night to Daytime on BDD100K

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Data preprocessing
  - Framework
  - Experiments

ACKNOWLEDGMENTS

The study is supported by the key research of Leshan Science and Technology Bureau Grant # A312022021.

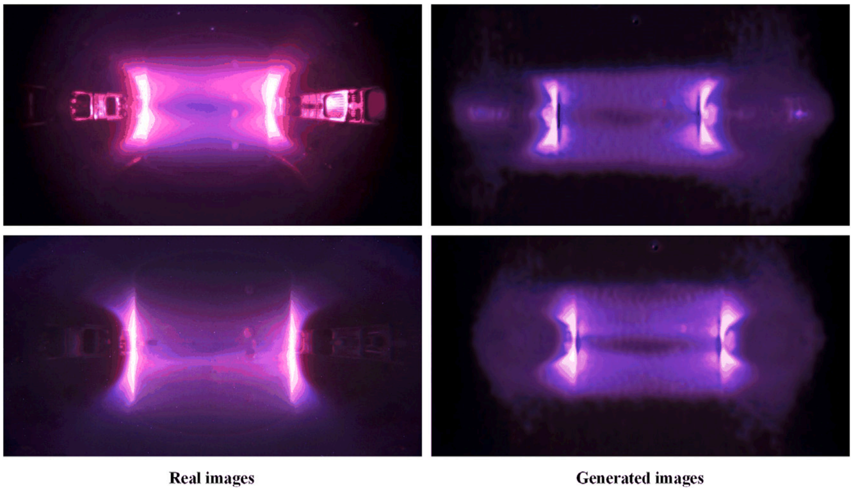


Figure 9. These cases are presented by the real plasma discharge images and images generated by CycleGAN

We found that the generated ones did not match the actual discharge process and many important details were missing.

AUTHOR CONTRIBUTIONS

The methods and framework were designed by J.Y.; all figures were designed and drawn by Z.L.G. and Jun J.Y.; all experimental data in the references were summarized by N.Y. and Q.W.; W.Y. conducted the final proofreading. All authors have read and agreed to the published version of the article.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: October 17, 2023
Revised: February 12, 2024
Accepted: March 26, 2024
Published: March 28, 2024

REFERENCES

1. Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Piscataway, NJ (IEEE), pp. 1440–1448.
2. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C. (2016). SSD: Single shot multibox detector. In ECCV (Berlin: Springer), pp. 21–37.
3. Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In CVPR (IEEE), pp. 779–788.
4. Redmon, J., and Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In CVPR (IEEE), pp. 7263–7271.
5. Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement. Preprint at arXiv. https://doi.org/10.48550/arXiv.1804.02767.
6. Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. NIPS (News Physiol. Sci.) *28*, 2969239–2969250.
7. Cao, X., Wang, Z., Yan, P., and Li, X. (2013). Transfer learning for pedestrian detection. Neurocomputing *100*, 51–57.
8. Saeidi, M., and Ahmadi, A. (2020). A novel approach for deep pedestrian detection based on changes in camera viewing angle. Signal Image Video Process. *14*, 1273–1281.
9. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. (2016). Domain separation networks. Adv. Neural Inf. Process. Syst. *29*.
10. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. J. Mach. Learn. Res. *17*, 1–35.
11. Gopalan, R., Li, R., and Chellappa, R. (2011). Domain adaptation for object recognition: An unsupervised approach. In International Conference on Computer Vision, Piscataway, NJ (IEEE), pp. 999–1006.
12. Long, M., Cao, Y., Wang, J., and Jordan, M.I. (2015). Learning transferable features with deep adaptation networks. In ICML (IMLS), pp. 97–105.
13. Long, M., Zhu, H., Wang, J., and Jordan, M.I. (2017). Deep transfer learning with joint adaptation networks. In ICML (IMLS), pp. 2208–2217.
14. Sun, B., and Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In ECCV (Berlin: Springer), pp. 443–450.
15. Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In CVPR (IEEE), pp. 7167–7176.
16. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. Preprint at arXiv. https://doi.org/10.48550/arXiv.1412.3474.
17. Hoffman, J., Wang, D., Yu, F., and Darrell, T. (2016). Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. Preprint at arXiv. https://doi.org/10.48550/arXiv.1612.02649.
18. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., and Chandraker, M. (2018). Learning to adapt structured output space for semantic segmentation. In CVPR (IEEE), pp. 7472–7481.
19. Chen, Y., Li, W., Sakaridis, C., Dai, D., and Van Gool, L. (2018). Domain adaptive faster R-CNN for object detection in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ (IEEE), pp. 3339–3348.
20. Hoffman, J., Guadarrama, S., Tzeng, E.S., Hu, R., Donahue, J., Girshick, R., and Saenko, K. (2014). LSDA: Large scale detection through adaptation. NIPS (News Physiol. Sci.) *27*, 3536–3545.
21. Inoue, N., Furuta, R., Yamasaki, T., and Aizawa, K. (2018). Cross-domain weakly-supervised object detection through progressive domain adaptation. In CVPR (IEEE), pp. 5001–5009.
22. Zhu, J.Y., Park, T., Isola, P., and Efros, A.A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In ICCV (IEEE), pp. 2223–2232.
23. Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ (IEEE), pp. 3354–3361.
24. Zhang, S., Benenson, R., and Schiele, B. (2017). Citypersons: A diverse dataset for pedestrian detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ (IEEE), pp. 3213–3221.
25. Fisher, Y., Wenqi, X., Yingying, C., Fangchen, L., Mike, L., and Vashisht, M. (2018). Bdd100k: a diverse driving video database with scalable annotation tooling. Preprint at arXiv. https://doi.org/10.48550/arXiv.1805.04687.
26. Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. *34*, 743–761.
27. Walk, S., Majer, N., Schindler, K., and Schiele, B. (2010). New features and insights for pedestrian detection. In CVPR (IEEE), pp. 1030–1037.
28. Benenson, R., Omran, M., Hosang, J., and Schiele, B. (2015). Ten years of pedestrian detection, what have we learned? In ECCV (Berlin: Springer), pp. 613–627.
29. Zhang, S., Benenson, R., and Schiele, B. (2015). Filtered channel features for pedestrian detection. CVPR *1*, 4. https://doi.org/10.48550/arXiv.1501.05759.
30. Hosang, J., Omran, M., Benenson, R., and Schiele, B. (2015). Taking a deeper look at pedestrians. In CVPR (IEEE), pp. 4073–4082.
31. Tian, Y., Luo, P., Wang, X., and Tang, X. (2015). Pedestrian detection aided by deep learning semantic tasks. In CVPR (IEEE), pp. 5079–5087.
32. Angelova, A., Krizhevsky, A., Vanhoucke, V., Ogale, A., and Ferguson, D. (2015). Real-time pedestrian detection with deep network cascades. In BMVC (British MVA).
33. Tian, Y., Luo, P., Wang, X., and Tang, X. (2015). Deep learning strong parts for pedestrian detection. In Proceedings of the IEEE International Conference on Computer VisionPiscataway, NJ (IEEE), pp. 1904–1912.
34. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., and Smeulders, A.W.M. (2013). Selective search for object recognition. Int. J. Comput. Vis. *104*, 154–171.
35. Lin, M., Li, C., Bu, X., Sun, M., Lin, C., Yan, J., and Deng, Z. (2020). Detr for crowd pedestrian detection. Preprint at arXiv. https://doi.org/10.48550/arXiv.2012.06785.
36. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., and Krishnan, D. (2017). Unsupervised pixel-level domain adaptation with generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ (IEEE), pp. 3722–3731.
37. Tsai, Y.H., Sohn, K., Schulter, S., and Chandraker, M. (2019). Domain adaptation for structured output via discriminative patch representations. In ICCV (IEEE), pp. 1456–1465.
38. Zhu, X., Pang, J., Yang, C., Shi, J., and Lin, D. (2019). Adapting object detectors via selective cross-domain alignment. In CVPR (IEEE), pp. 687–696.

39. Saito, K., Ushiku, Y., Harada, T., and Saenko, K. (2019). Strong-weak distribution alignment for adaptive object detection. In CVPR (IEEE), pp. 6956–6965.

40. Kim, T., Jeong, M., Kim, S., Choi, S., and Kim, C. (2019). Diversify and match: A domain adaptive representation learning paradigm for object detection. In CVPR (IEEE), pp. 12456–12465.

41. Cai, Z., and Vasconcelos, N. (2021). Cascade R-CNN: High quality object detection and instance segmentation. IEEE Trans. Pattern Anal. Mach. Intell. *43*, 1483–1498.

42. Ganin, Y., and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In International Conference on Machine Learning (IMLS), pp. 1180–1189.

43. Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. JMLR *9*, 2579–2605.

44. Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. JMLR, 315–323.

45. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In CVPR (IEEE), pp. 3213–3223.

46. Hasan, I., Liao, S., Li, J., Akram, S.U., and Shao, L. (2021). Generalizable Pedestrian Detection: The Elephant In The Room. In CVPR (IEEE), pp. 11323–11332.

47. Shen, G., Yu, Y., Tang, Z.R., Chen, H., and Zhou, Z. (2022). HQA-Trans: An end-to-end high-quality-awareness image translation framework for unsupervised cross-domain pedestrian detection. IET Comput. Vis. *16*, 218–229.

48. Jiao, Y., Yao, H., and Xu, C. (2021). SAN: Selective alignment network for cross-domain pedestrian detection. IEEE Trans. Image Process. *30*, 2155–2167.

49. Schulz, D., and Perez, C.A. (2023). Two-stage pedestrian detection model using a new classification head for domain generalization. Sensors *23*, 9380.

50. Zhang, S., Xie, Y., Wan, J., Xia, H., Li, S.Z., and Guo, G. (2020). Widerperson: A diverse dataset for dense pedestrian detection in the wild. IEEE Trans. Multimed. *22*, 380–393.

51. Wang, H., Liao, S., and Shao, L. (2021). Afan: Augmented feature alignment network for cross-domain object detection. IEEE Trans. Image Process. *30*, 4046–4056.

52. Wang, X., Huang, T.E., Liu, B., Yu, F., Wang, X., Gonzalez, J.E., and Darrell, T. (2021). Robust object detection via instance-level temporal cycle confusion. ICCV *8*.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Citypersons	Zhang et al.[24]	https://www.cityscapes-dataset.com/downloads
BDD100k	Fisher et al.[25]	http://bdd-data.berkeley.edu/download.html
KITTI	Geiger et al.[23]	https://www.cvlibs.net/datasets/kitti/index.php
Caltech	Dollar et al.[26]	http://www.vision.caltech.edu/datasets
Foggy Cityscapes	Cordts et al.[45]	https://www.cityscapes-dataset.com/downloads
WiderPedestrian	Zhang et al.[50]	https://competitions.codalab.org/competitions/20132
Software and algorithms		
GRL	Ganin et al.[42]	https://github.com/fungtion/DANN
CycleGAN	Zhu et al.[22]	https://github.com/open-mmlab/mmgeneration
Cascade R-CNN	Cai et al.[41]	https://github.com/open-mmlab/mmdetection
t-SNE	Van der Maatenet al.[43]	https://github.com/KlugerLab/pyFIt-SNE
Pedestrian detection	This paper	https://github.com/haoqiu111/MY-LEARNING

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Jun Yao (40516473@qq.com)

Materials availability

This study did not generate any new physical materials.

Data and code availability

The main model of this paper is an open-source model, available at https://github.com/haoqiu111/MY-LEARNING.

Data and code needed to reproduce the research and figures presented in this study are fully documented and accessible in the key resources table. Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Data preprocessing

Data preprocessing includes resizing, normalization, and augmentation. To meet the requirements of our proposed cross-domain detection framework, it is necessary to adjust the images of the training and testing sets to the size of (3,224,224) in all the datasets mentioned in the previous sections.

In addition, set the MEAN and STD values of the normalization options to [0.471, 0.448, 0.408] and [0.234, 0.239, 0.242], which can eliminate stable distributions in images and highlight the differences and features of each image. The intensity of the original images should be scaled to [0,1] instead of [0.255]. Normalization will subtract MEAN and divide by STD, as shown in Equation 6.

$$input\_image = \frac{raw\_image/255 - \mathrm{MEAN}}{\mathrm{STD}} \quad \text{(Equation 6)}$$

Augmentation options were set during the training phase of all datasets, such as random brightness contrast, horizontal flip, rotate, crop, and random gamma. All the augmentation options have been proved by experiments to significantly improve the effectiveness of cross-domain detection models and reduce the MR.

Framework

The pedestrian detection and image generation models both use the OpenMMlab framework which includes MMDetection and MMGeneration projects, making it extremely convenient to train the models in this study.

In MMDetection, we use the Cascade R-CNN configuration file and add a domain feature-aligned branch according to experimental requirements to train a domain discriminator with adversarial learning loss functions.

In MMGeneration, we use the CycleGAN configuration file to train a CycleGAN model for generating images of intermediate domains in each cross-domain task.

Experiments

We use the Adam optimizer to train cross-domain detection models, with a hyper-parameter set to 0.9 (first moment estimation, β1) and 0.999 (second moment estimation, β2). They used CosineAnnealing for the Lr-attenuation (learning rate attenuation strategy). The learning rate in CosineAnnealing first slowly decreases as the cosine value increases, then accelerates the decrease, and then slowly decreases again. This descent mode produces good training results with a highly effective calculation method.

In detection tasks, we used a Cascade R-CNN model pre-trained on the COCO dataset, with training epochs of 200, 300, and 500 on cross-domain tasks KITTI → Citypersons, Citypersons → BDD100k, and BDD100k → Citypersons, respectively. When training, the batch size is usually set to 8, and the loss functions are focal and GIOU.

In image generation tasks, we need to specify a root directory and then store the source domain images in the folders train_A and test_A. Store the target domain images in the folder train_B and test_B. Then we set Buffer_ Size = 50, and randomly sample 50 images to participate in each update. The loss function consists of 2 cycle_loss and 2 id_loss with weights of 10 and 0.5. We use the Adam optimizer, where the moment estimates for both stages are set to β1 = 0.5, β2 = 0.999. The image normalization standard is MEAN = [0.5,0.5,0.5], STD = [0.5,0,0.5]. The evaluation metrics for the generator are FID and IS. The total number of training epochs is 80000.