Six-Channel Image Representation for Cross-Domain Object Detection

Tianxiao Zhang[1], Wenchi Ma[1], and Guanghui Wang[2](✉)

[1] Department of Electrical Engineering and Computer Science, University of Kansas,
Lawrence, KS 66045, USA
[2] Department of Computer Science, Ryerson University,
Toronto, ON M5B 2K3, Canada
wangcs@ryerson.ca

Abstract. Most deep learning models are data-driven and the excellent performance is highly dependent on the abundant and diverse datasets. However, it is very hard to obtain and label the datasets of some specific scenes or applications. If we train the detector using the data from one domain, it cannot perform well on the data from another domain due to domain shift, which is one of the big challenges of most object detection models. To address this issue, some image-to-image translation techniques have been employed to generate some fake data of some specific scenes to train the models. With the advent of Generative Adversarial Networks (GANs), we could realize unsupervised image-to-image translation in both directions from a source to a target domain and from the target to the source domain. In this study, we report a new approach to making use of the generated images. We propose to concatenate the original 3-channel images and their corresponding GAN-generated fake images to form 6-channel representations of the dataset, hoping to address the domain shift problem while exploiting the success of available detection models. The idea of augmented data representation may inspire further study on object detection and other applications.

Keywords: Object detection · Domain shift · Unsupervised image-to-image translation

1 Introduction

Computer vision has progressed rapidly with deep learning techniques and more advanced and accurate models for object detection, image classification, image segmentation, pose estimation, and tracking emerging almost every day [31,42,49]. Even though computer vision enters a new era with deep learning, there are still plenty of problems unsolved and domain shift is one of them. Albeit CNN models are dominating the computer vision, their performances often become inferior when testing some unseen data or data from a different domain, which is denoted as domain shift. Since most deep learning models are data-driven and

Y. Peng et al. (Eds.): ICIG 2021, LNCS 12888, pp. 171–184, 2021.
https://doi.org/10.1007/978-3-030-87355-4_15

the high-accurate performance is mostly guaranteed by the enormous amount of various data, domain shift often exists when there are not enough labeled specific data but we have to test those kinds of data in the testing set. For instance, although we only detect cars on the roads, training the models on day scenes cannot guarantee an effective detection of cars in the night scenes. We might have to utilize enough datasets from night scenes to train the models, nonetheless, sometimes the datasets from some specific scenes are rare or unlabeled, which makes it even more difficult to mitigate the domain shift effect.

To mitigate the situation where some kinds of training data are none or rare, The image-to-image translation that could translate images from one domain to another is highly desirable. Fortunately, with the advent of Generative Adversarial Networks (GANs) [15], Some researchers aim to generate some fake datasets in specific scenes using GAN models to overcome the lack of data. With some unpaired image-to-image translation GAN models (i.e., CycleGAN [52]), it can not only translate images from the source domain to target domain, but also translate images from target domain to source domain, and the entire process does not require any paired images, which make it ideal for real-world applications.

The GAN models for image-to-image translation can generate the corresponding fake images of the target domain from the original images of the source domain in the training dataset, and we can utilize the GAN-generated images to train object detection models and test on images of target domain [2]. Since we expect to solve cross-domain object detection problems, after pre-processing the data and generating the fake images with image-to-image translation models, the generated data has to be fed into the object detection models to train the model and the trained model could demonstrate its effectiveness through testing the data from the target domain. Employing GAN-generated fake images to train the detection models to guarantee the domain of the training data and testing data being the same illustrated the effectiveness of the approach and the detection performance was boosted for the scenario where the training data for the detection models is from one domain while the testing data is in another domain [2].

Instead of simply utilizing the fake images to train the model, we propose to solve the problem from a new perspective by concatenating the original images and their corresponding GAN-translated fake images to form new 6-channel representations. For instance, if we only have source domain images but we intend to test our model on unlabeled images in the target domain, what we did was training the image-to-image translation model with source domain data and target domain data. And then we could employ the trained image translation model to generate the corresponding fake images. Since some image-to-image translation models [52] could translate images in both directions, we are able to acquire the corresponding fake data for the data from both the source domain and target domain. Thus, both training images and testing images would be augmented into 6-channel representations by concatenating the RGB three channels of the original images with those from the corresponding fake images. Then we can train and test the detection models using available detection models, the only difference is the dimension of the kernel of the CNN models for detection in the first layer becomes 6 instead of 3. The process of training and testing the proposed method is depicted in Fig. 1.

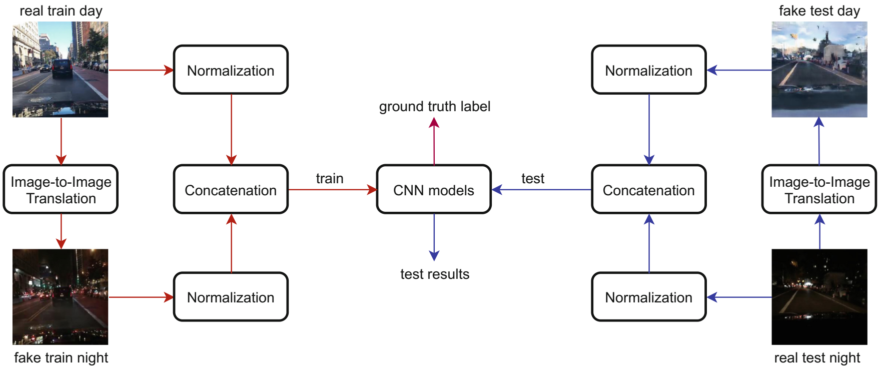

Fig. 1. The flow chart of the proposed 6-channel image augmentation approach for training and testing CNN-based detection models.

2 Related Work

Image-to-image translation is a popular topic in computer vision [43,44]. With the advent of Generative Adversarial Networks [15], it could be mainly categorized as supervised image-to-image translation and unsupervised image-to-image translation [1]. The supervised image-to-image translation models such as pix2pix [20] and BicycleGAN [54], require image pairs from two or more domains (i.e., the exact same image scenes from day and night), which are extremely expensive and unrealistic to be acquired in the real world. Perhaps the quality of the translated images is sometimes beyond expectations, they are not ideal for real-world applications.

The unsupervised image-to-image translation models can be divided as cycle consistency based models (i.e., CycleGAN [52], DiscoGAN [22], DualGAN [46]) which introduce cycle consistency losses, autoencoder based models (i.e., UNIT [29]) combined with autoencoder [23], and recent disentangled representation models (i.e., MUNIT [18], DIRT [25]). Since the unsupervised image-to-image translation models only require image sets from two or more domains and do not necessitate any paired images which are arduous to collect and annotate, they are often leveraged to generate some fake data in the target domain and applied to other computer vision tasks such as object detection and image classification. Among those unsupervised image-to-image translation models, CycleGAN [52] is frequently utilized as the image-mapping model to generate some fake data to be employed in some cross-domain problems [2,19].

Object detection addresses the problem that detects the semantic instances on digital images or videos. The fundamental purpose of object detection is to classify the objects shown on the images or videos and simultaneously locate those objects by coordinates [32]. The applications of object detection are in various fields such as medical image analysis [33], self-driving car, pose estimation, segmentation, etc.

From the perspective of stages, the object detectors are categorized into two types: one-stage detectors and two-stage detectors. For two-stage object detectors such as Faster R-CNN [37], MS-CNN [4], R-FCN [8], FPN [26], these models are often comprised of a region proposal network as the first stage that selects the candidate anchors which have high probabilities to contain objects and a detection network as the second stage that classify the objects to be contained by these candidates and further do the bounding box regression for these candidates to refine their coordinates and finally output the results. For one-stage object detectors like SSD [30], YOLOv1-v4 [3,34–36], RetinaNet [27], these detectors often directly classify and regress the pre-defined anchor boxes instead of choosing some candidates. Thus the two-stage models often outperform the one-stage counterparts while one-stage models frequently have a faster inference rate than two-stage approaches.

Due to the various sizes and shapes of the objects, some models [26,27,30,48] design anchor boxes on different levels of feature maps (the pixels on lower level feature maps have a small receptive field and the pixels on higher-level feature maps have large receptive field) so that the anchors on lower level features are responsible for the relative small objects and the anchors on higher-level features are in charge of detecting relatively large objects. The middle-sized objects are perhaps recognized by the middle-level feature maps.

The aforementioned detection models are anchor-based that we have to design pre-defined anchor boxes for these models. In recent years, some anchor-free models [10,24,39,50,51] are attracting great attention for their excellent performance without any pre-defined anchor boxes. Some of them are even dominating the accuracy on COCO benchmark [28]. Since a large amount of anchors has to be generated for some anchor-based models and most of them are useless because no object is contained in the majority of anchors, anchor-free models might predominate in the designs of object detectors in the future. Recently, the transformer [40] is applied successfully to object detection [5], which is an anchor-free model with attention mechanisms.

Nonetheless, many problems have not been well solved in this field, especially in cross-domain object detection. Since modern object detectors are based on deep learning techniques and deep learning is data-driven so that the performance of modern object detectors is highly dependent on how many annotated data can be employed as the training set. Cross-domain issues arise when there are not enough labeled training data that have the same domain as the testing data, or the dataset is diverse or composed of various datasets of different domains in both training and testing data.

Domain Adaptive Faster R-CNN [6] explores the cross-domain object detection problem based on Faster R-CNN. By utilizing Gradient Reverse Layer (GRL) [12] in an adversarial training manner which is similar to Generative Adversarial Networks (GAN)[15], this paper proposes an image-level adaptation component and an instance-level adaptation component which augment the Faster R-CNN structure to realize domain adaptation. In addition, a consistent regularizer between those two components is to alleviate the effects of the domain shift between different dataset such as KITTI [13], Cityscapes [7], Foggy Cityscapes [38], and SIM10K [21].

Universal object detection by domain attention [41] addresses the universal object detection of various datasets by attention mechanism [40]. The universal object detection is arduous to realize since the object detection datasets are diverse and there exists a domain shift between them. The paper [17] proposes a domain adaption module which is comprised of a universal SE adapter bank and a new domain-attention mechanism to realize universal object detection. [19] deals with cross-domain object detection that instance-level annotations are accessible in the source domain while only image-level labels are available in the target domain. The authors exploit an unpaired image-to-image translation model (CycleGAN [52]) to generate fake data in the target domain to fine-tune the trained model which is trained on the data in the source domain. Finally, the model is fine-tuned again on the detected results of the testing data (pseudo-labeling) to make the model even better.

The study [2] utilizes CycleGAN [52] as the image-to-image translation model to translate the images in both directions. The model trained on the fake data in the target domain has better performance than that trained on the original data in the source domain on testing the test data from the target domain. The dataset we employ in this paper is from [2] and we follow exactly the same preprocessing procedure to prepare the dataset. In the following, we will discuss our proposal that utilizes concatenated image pairs (real images and corresponding fake images) to train the detection model and compare it to the corresponding approach from [2].

3 Proposed Approach

The framework of our proposed method is depicted in Fig. 1. In our implementation, we employ CycleGAN for image-to-image translation, which is trained with the data from the source domain (i.e., day images) and the data from the target domain (i.e., night images). First, the fake data (target domain) is generated from the original data (source domain) via the trained image-to-image translation model (i.e., generating the fake night images from the real day images). Then, the real and fake images are normalized and concatenated (i.e., concatenating two 3-channel images to form a 6-channel representation of the image). Finally, the concatenated images are exploited to train the CNN models. During the stage of test, the test data is processed in a similar way as the training data to form concatenated images and sent to the trained CNN model for detection.

3.1 Image-to-Image Translation

To realize the cross-domain object detection, we have to collect and annotate the data in the target domain to train the model. While it is difficult to acquire the annotated data in the target domain, image-to-image translation models provide an option to generate fake data in the target domain.

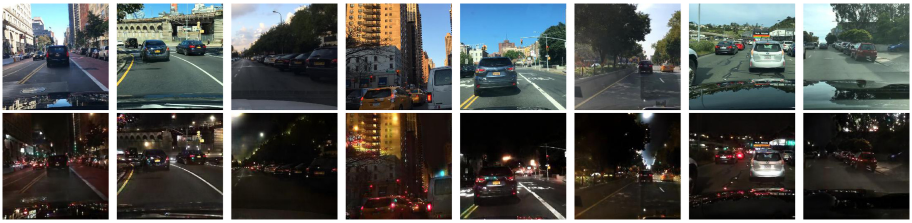

Fig. 2. Several samples of original-day images (1st row) and their corresponding GAN-generated fake-night images (2nd row).

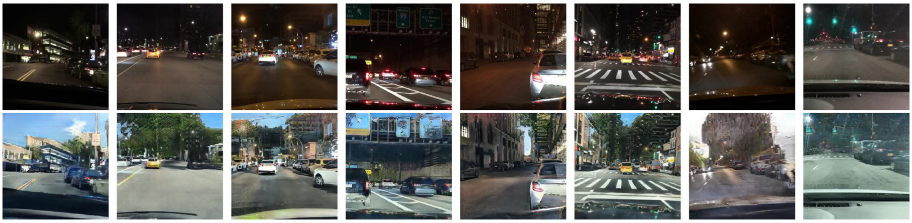

Fig. 3. Several samples of original-night images (1st row) and their corresponding GAN-generated fake-day images (2nd row).

In our experiment, we employed an unpaired image-to-image translation model: CycleGAN [52]. CycleGAN is an unsupervised image-to-image translation that only requires images from two different domains (without any image-level or instance-level annotations) to train the model. Furthermore, unpaired translation illustrates that the images from two domains do not need to be paired which is extremely demanding to be obtained. Last but not least, the locations and sizes of the objects on the images should be the same after the image-to-image translation so that any image-level labels and instance-level annotations of the original images can be utilized directly on the translated images. This property is extraordinarily significant since most CNN models are data-driven and the annotations of the images are indispensable to successfully train the supervised CNN models (i.e., most object detection models). Unpaired image-to-image translation models such as CycleGAN [52] can translate the images in two directions without changing the key properties of the objects on the images. Thus the annotations such as coordinates and class labels of the objects on the original images can be smoothly exploited in the fake translated images. As manually annotating the images is significantly expensive, by image-to-image translation, the translated images would automatically have the same labels as their original counterparts, which to some extent makes manually annotating images unnecessary.

3.2 CNN Models

In Fig. 1, the CNN model can be any CNN-based object detection model, where the dimension of the convolutional kernel in the first layer is changed from 3 to 6. In our implementation, we employ Faster R-CNN [37] for detection, and we use ResNet-101 [16] as the backbone network for the detection model.

Faster R-CNN is a classic two-stage anchor-based object detector that is comprised of Region Proposal Network (RPN) and detection network. Since it is an anchor-based model, we have to design some pre-defined anchor boxes on the feature maps. Typically, 9 anchors with 3 different sizes and 3 different aspect ratios are designed to act as the pre-defined anchor boxes on each location of the feature maps. The objective of RPN is to select some region proposals with a high probability of containing objects from the pre-defined anchors and further refine their coordinates. Each pre-defined anchor would be associated with a score indicating the probability of that anchor box containing an object. Only the anchor boxes with associated scores higher than some threshold can be selected as region proposals and those region proposals are further refined by RPN and later fed into the detection network.

The purpose of the detection network is to receive the region proposals selected and refined by RPN and finally do the classification for each rectangle proposal and bounding box regression to improve the coordinates of the box proposals. Since the region proposals may have various sizes and shapes, more accurately, the number of elements each proposal has might be varying. To guarantee the region proposals are fed into the fully connected layers effectively (the fully connected layer needs the length of input data fixed), the ROI pooling layer is adopted to ensure the size of the input of each proposal to the detection network is fixed. The detection network is simply from Fast R-CNN [14] that is to classify the object which might be contained by each region proposal and simultaneously refine the coordinates of the rectangle boxes. The output of the Faster R-CNN network is the class of the object each proposal might include and the coordinates of the bounding box for each refined proposal.

4 Experiments

In this section, the datasets and the experimental methodology and parameter settings are elaborated. We conducted some of the experiments from [2] for comparison.

4.1 Datasets

We employ the same dataset as [2] in our experiments. The original datasets are from BDD100K [47] which is a large-scale diverse dataset for driving scenes. Since the dataset is extremely large and contains high-resolution images and various scenarios on the road and the weather conditions (sunny, rainy, foggy, etc.) [2], the authors only choose the clear or partly cloudy day and night images to demonstrate the domain shift from day to night [2]. In addition, all selected images are cropped to 256 × 256 pixels with proper adjustment. There are a total 12,000 images left and processed (6,000 day images and 6,000 night images). After that, the images are randomly sampled and divided into four sets: train-day, train-night, test-day, and test-night, each of the sets contains 3,000 256 × 256 images. We harness the set of train-day and train-night to train the CycleGAN model and utilized the trained GAN model to generate fake train-night (from train-day), fake train-day (from train-night), fake test-night (from test-day), and fake test-day (from test-night). Now we have a total of 12,000 real images (3,000 for each set) and 12,000 fake images (3,000 for each set). Then we can concatenate the real images and their corresponding fake images to generate 6-channel representations that would be fed into the Faster R-CNN object detector. After choosing and processing the images, the car is the only object on the image to be detected. Some samples of real images and their corresponding GAN-generated fake counterparts are illustrated in Fig. 2 and Fig. 3.

4.2 Experimental Evaluations

Faster R-CNN model is implemented in Python [45] with Pytorch 1.0.0 and CycleGAN is implemented in Python [53] with PyTorch 1.4.0. All experiments are executed with CUDA 9.1.85 and cuDNN 7 on a single NVIDIA TITAN XP GPU with a memory of 12 GB.

The metric we employed is mean Average Precision (mAP) from PASCAL VOC [11], which is the same metric employed in [2]. Since the car is the only object to be detected, the mAP is equivalent to AP in this dataset since mAP calculating the mean AP for all classes.

For CycleGAN, the parameters are default values in [53]. For Faster R-CNN, similarly to [2], we utilize pre-trained ResNet-101 [16] on ImageNet [9] as our backbone network. We select the initial learning rates from 0.001 to 0.00001 and the experiments are implemented separately for those chosen initial learning rates, but we do not utilize them all for each experiment since our experiments demonstrate that the higher the learning rate we selected from above, the better the results would be. In each 5 epoch, the learning rate decays as 0.1 of the previous learning rate. The training process would be executed 20 to 30 epochs, but the results indicate that the Faster R-CNN model converges relatively early on the dataset. Training every 5 epochs, we record the testing results on test data, but we would report the best one for each experiment. The model parameters are the same for 6-channel experiments and 3-channel experiments, except for 6-channel experiments, the kernel dimension of the first layer of the Faster R-CNN model is 6 instead of 3. And we just concatenate each kernel by itself to create 6-dimension kernels in the first layer of ResNet-101 backbone for 6-channel experiments. While for 3-channel experiments, we simply exploit the original ResNet-101 backbone as our initial training parameters.

4.3 Experimental Results

First, we implemented the training and testing of the original 3-channel Faster R-CNN model which is illustrated in Table 1. The test set is test-night data which is fixed. With different training sets, the detection results on test night are varying.

Table 1. 3-channel detection

Train set	mAP
Train-day (3,000 images)	0.777
Fake train-night (3,000 images)	0.893
Train-night (3,000 images)	0.933
Train-day + train-night (6,000 images)	0.941

From Table 1 we can see that, for testing the test-night set, the model trained on the fake-train night set is much better than that trained on the original train-day set, which corresponds to the results from [2]. These experimental results indicate that if the annotated day images are the only available training data while the test set contains only night images, we could leverage fake night images generated by the image-to-image translation models to train the CNN model. The results are excellent when the model is trained on the train-night set (without domain shift), indicating the domain shift is the most significant influence on the performance of the CNN model in this experiment.

Then we conduct the experiments for our proposed 6-channel Faster R-CNN model which is shown in Table 2. The test data is comprised of test-night images concatenated with corresponding translated fake test-day images. The training sets in Table 2 have 6 channels. For instance, train-day in the table indicates train-day images concatenated with corresponding fake train-night images, and train-day plus train-night in the table represents train-day images concatenated with corresponding fake train-night images plus train-night images concatenated with corresponding fake train-day images.

Table 2. 6-channel detection

Train set	mAP
Train-day (3,000 6-channel representations)	0.830
Train-night (3,000 6-channel representations)	0.931
Train-day + train-night (6,000 6-channel representations)	0.938

From Table 1 and Table 2, it is noticeable that even though the model trained on train-day images concatenated with fake train-night images (6-channel) has a better result with AP 0.830 than that just training on train-day (3-channel) with AP 0.777, it is worse than the model only trained on fake train-night (3-channel) with AP 0.893.

To demonstrate if the 6-channel approach can improve the detection results in the situation where the training set and testing set do not have domain shift, we also performed the experiment that trains the model on train-night set (3-channel) and tests it on test-night set. From Table 1, the average precision is 0.933, which is pretty high since there is no domain shift between the training data and testing data. Accordingly, we did the corresponding 6-channel experiment which trains on train-night set concatenated with fake train-day set and tests it on test-night images concatenated with fake test-day images. From Table 2, the average precision of this 6-channel model is almost the same as its corresponding 3-channel model.

We increase the size of the training data by training the model with the train-day set plus the train-night set and testing it on test-night data. From Table 1 and Table 2, the result of 6-channel model also performs similar to its 3-channel counterpart. More experimental results are shown in Table 3, which are from the original 3-channel models. To remove the effect of domain shift, the training set and the testing set do not have domain shift (they are all day images or night images). From Table 3, it is obvious that the "quality" shift influences the performance of the models. For instance, the model trained on the original train-day (or train-night) set has better performance on the original test-day (or test-night) set than the GAN-generated fake day (or night) images. Similarly, the model which is trained on GAN-generated fake train-day (or fake train-night) set performs better on the GAN-generated fake test-day (or fake test-night) set than the original test-day (or test-night) set.

Table 3. 3-channel extra experiments

Train set	Test set	mAP
Train-day	Test-day	0.945
	Fake test-day	0.789
Fake train-day	Fake test-day	0.914
	Test-day	0.903
Train-night	Test-night	0.932
	Fake test-night	0.859
Fake train-night	Fake test-night	0.924
	Test-night	0.868

5 Conclusion

The study has evaluated a 6-channel approach to address the domain-shift issue by incorporating the generated fake images using image-to-image translation.

However, we have not achieved the expected results. One possible reason is the quality of the generated images is inferior compared to the original images, especially the fake day images generated from the data of night scenes, as illustrated in Fig. 2 and Fig. 3. If we merely concatenate the original high-quality images with their inferior counterparts, the model may treat the low-quality fake image channels as some kind of "noise", and thus, the model could hardly learn more useful information from the concatenated 6-channel representations. Another possible reason is that the domain shift issue may still exist in the combined 6-channel representations, which prevents the model from extracting useful information from the concatenated representations. Moreover, the dataset we used in the experiments only has limited samples, which are insufficient to train the model. We hope the idea of augmented data representation can inspire more further investigations and applications.

References

1. Alotaibi, A.: Deep generative adversarial networks for image-to-image translation: a review. Symmetry **12**(10), 1705 (2020)
2. Arruda, V.F., et al.: Cross-domain car detection using unsupervised image-to-image translation: from day to night. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2019)
3. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
4. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part IV. LNCS, vol. 9908, pp. 354–370. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_22
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. arXiv preprint arXiv:2005.12872 (2020)
6. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster R-CNN for object detection in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3339–3348 (2018)
7. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)
8. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. arXiv preprint arXiv:1605.06409 (2016)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
10. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: keypoint triplets for object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6569–6578 (2019)
11. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2015)
12. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180–1189. PMLR (2015)

13. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
14. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
15. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
17. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
18. Huang, X., Liu, M.-Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018, Part III. LNCS, vol. 11207, pp. 179–196. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_11
19. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5001–5009 (2018)
20. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
21. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R.: Driving in the matrix: can virtual worlds replace human-generated annotations for real world tasks? arXiv preprint arXiv:1610.01983 (2016)
22. Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192 (2017)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
24. Law, H., Deng, J.: Cornernet: detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 734–750 (2018)
25. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: Proceedings of the European Conference on computer vision (ECCV), pp. 35–51 (2018)
26. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
27. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
28. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
29. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in Neural Information Processing Systems, pp. 700–708 (2017)
30. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part I. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2

31. Ma, W., Li, K., Wang, G.: Location-aware box reasoning for anchor-based single-shot object detection. IEEE Access **8**, 129300–129309 (2020)
32. Ma, W., Wu, Y., Cen, F., Wang, G.: MDFN: multi-scale deep feature learning network for object detection. Pattern Recognit. **100**, 107149 (2020)
33. Mo, X., Tao, K., Wang, Q., Wang, G.: An efficient approach for polyps detection in endoscopic videos based on faster R-CNN. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3929–3934. IEEE (2018)
34. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
35. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
36. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
37. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
38. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. Int. J. Comput. Vis. **126**(9), 973–992 (2018)
39. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: fully convolutional one-stage object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 9627–9636 (2019)
40. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
41. Wang, X., Cai, Z., Gao, D., Vasconcelos, N.: Towards universal object detection by domain attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7289–7298 (2019)
42. Wu, Y., Zhang, Z., Wang, G.: Unsupervised deep feature transfer for low resolution image classification. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (2019)
43. Xu, W., Keshmiri, S., Wang, G.: Stacked wasserstein autoencoder. Neurocomputing **363**, 195–204 (2019)
44. Xu, W., Shawn, K., Wang, G.: Toward learning a unified many-to-many mapping for diverse image translation. Pattern Recognit. **93**, 570–580 (2019)
45. Yang, J., Lu, J., Batra, D., Parikh, D.: A faster Pytorch implementation of faster R-CNN (2017). https://github.com/jwyang/faster-rcnn.pytorch
46. Yi, Z., Zhang, H., Tan, P., Gong, M.: DualGAN: unsupervised dual learning for image-to-image translation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2849–2857 (2017)
47. Yu, F., et al.: Bdd100k: a diverse driving video database with scalable annotation tooling. arXiv preprint arXiv:1805.04687 2(5), 6 (2018)
48. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4203–4212 (2018)
49. Zhang, X., Zhang, T., Yang, Y., Wang, Z., Wang, G.: Real-time golf ball detection and tracking based on convolutional neural networks. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2808–2813. IEEE (2020)
50. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 850–859 (2019)

51. Zhu, C., He, Y., Savvides, M.: Feature selective anchor-free module for single-shot object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 840–849 (2019)
52. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)
53. Zhu, J.Y., Park, T., Wang, T.: CycleGAN and pix2pix in Pytorch (2020). https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix
54. Zhu, J.Y., et al.: Toward multimodal image-to-image translation. In: Advances in Neural Information Processing Systems, pp. 465–476 (2017)