Common Space Object Detection For Domain Adaptation

Vikash Kumar
Department of Computational and Data Sciences
Indian Institute Of Science
Bangalore, 560012
vikashks@iisc.ac.in

Abstract

For training a deep learning based detection model, we need huge labelled dataset.Labelling is a tedious task. Often model trained in one domain fails for generalize well in other domain for similar task at hand.It happens due to the mismatch of data distribution between domains. Domain adaptation tries to minimize the domain gap between distributions of source domain and target domain.We don't need label information for target domain and hence we avoid tedious task of labelling. In Common Space Object Detection For Domain Adaptation, we bring source and target domain distribution closer by generating synthetic samples using CycleGAN which only requires image level annotations. We train an adaptation network using adversarial loss to generate domain invariant features. Additionally, I also explored pseudo labelling for fine-tuning the trained model by adaptation network.

1 Introduction

Object detection is one the most import and challenging field of study in computer vision. We perform object classification and localization of object in an image frame of sequence of image frames.Due to improvement and availability of computational power and advances in in deep learning algorithms, object detection algorithm performance has improved significantly[5, 18, 15, 16, 17, 13]. Most of the works which shows promising results are done in supervised setting where we need a lot of annotated training data. Annotation includes polygon bounding box co-ordinates and class information. Most of the data we collect are in raw form and annotating is time consuming and costly. In many application, it is not even possible to get the annotation. For instance, autonomous vehicles need to detect object in all weather condition but its not possible to include samples from all the weather condition. Similarly, if we want to deploy our autonomous vehicle object detection model to different city or different country, its performance if going to be heavily affected due to the presence of data biases.

Domain adaptation method in which we don't use any annotation or label information in target domain is popularly called Unsupervised domain adaptation (UDA).Data distribution of source domain and target domain are different i.e $P_s \neq P_t$ where P_s is source data distribution and P_t is target data distribution. Our ultimate goal is to minimize the domain gap between source and target domain so that applications built using source dataset can also be deployed in target domain where we don't have annotation. Object detection for domain adaptation is relatively harder task compared to object classification because domain gap is large in the previous case.It also makes progress in object detection for domain adaptation slower compared to classification task. In this paper, we try to align source and target distribution in the latent domain with the help of an encoder which get connected to a detection network and a discriminator network as suggested in [3]. We don't do it in original image space instead we translate source domain images near to the target using image to

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

translation network[14] which generates synthetic images having content of source image but style of target image.Generated source become our new source and then we translate target near to new source. Finally adaptation happens between new source and new target. After training adaptation network, we fine-tune the trained model using pseudo labelling [9] which we generate using trained model and taking the bounding boxes corresponding to maximum score across all classes which are greater than the threshold.

2 Related Work

2.1 Object Detection

Recent Deep learning based object detection algorithm which have achieved the state of the art performance uses deep convolutional neural network (CNN) for feature extraction.Fast-RCNN [5] used selective search [22] and Faster-RCNN [18] uses region proposal network. Faster-RCNN is very fast compared to Fast-RCNN due to the presence of region proposal network as a substitute for selective search. Other Object detection algorithms such as YOLO [17], SSD [13] and RetinaNet [12] are based on single shot which reduces the computation time further. We have used Faster-RCNN network for our implementation.

All these methods are implemented in supervised setting so we need the data annotation for training these network. Dependency on annotation for achieving the state of the art performance make them useful only in those situations when have huge amount of annotated training data. Our UDA method doesn't having annotation for target domain and it uses progressive training and pseudo labelling methods to achieve the detection objecting in the wild.

2.2 Image Generation

Generative Adversarial Network(GAN) [7] have played an important role in the recent advances of computer vision. It is based on two player zero-sum game. Generator and discriminator network are two players who compete with each-other and equilibrium between generator and discriminator loss is achieved.

Paired image to image translation[10] and unpaired image to image translation [14] are two widely image translation methods.Paired image to translation requires mapping of one image to another for training and useful in applications like sketch to natural image generation.We don't have corresponding pair of images available.So We have used CycyleGAN [14] in our method which helps us to generate synthetic samples closer to the target domain. It is having two generator and two discriminator. It uses cycle consistency loss to ensure that mode collapse doesn't occur and generated images are more like natural images.

2.3 Weakly Supervised Object Detection

Weakly supervised detector(WSD) are used to annotate dataset using a pre-trained detection network. In our current work, we extract the region proposal using a trained model in common space as explained in the proposed method section. It returns many bounding boxes per image. We select bounding box corresponding to the best detection score which is more than threshold. These methods aren't very accurate but helps during fine-tuning. other approaches are based on two-stream CNN [1, 11, 21] where two outputs are generated per image and then train the network to minimize the difference of output as minimization of L2 loss.

2.4 Domain Adaptation

Domain adaptation tries to minimize the domain gap between source and target data distribution. Source domain dataset are annotated dataset whereas target domain dataset are not annotated. Search for domain invariant representation have been explored using conventional techniques such as Geodesic Flow Kernel(GFK) [6]. Adversarial training has shown significant improvement in both object detection and object classification.

Domain Adverserail Neural Network(DANN) [3] was one of major break through in domain adaptation. DANN architecture consist of encoder network, discriminator network, and detection network.

Encoder gives domain invariant feature after training this network. Detection network is trained using Source domain dataset with there annotations. Discriminator is trained using source and target dataset where target dataset acts as fake generated data and source dataset as the real and natural dataset. Negative gradient flows between encoder network and discriminator network. To minimize the domain gap before applying adaptation we first bring the image space closer using unpaired image to image translation network.It makes the adaptation task easier as the domain gap is reduced in the new setting.

3 Proposed Method

3.1 Problem Scenario and Proposed Architecture

For solving the domain adaptation problem in object detection, we propose to perform the adaptation in synthetic image space followed by fine-tuning with the help of weakly supervised detection.Synthetic images are generated such that it minimizes the domain gap between source and target domain. We represent the source, synthetic source, target and synthetic target as S, S_s, T, T_s respectively.Usually adaptation happens between source domain and target domain are represented as $S \rightarrow T$. In current setting S_s and S_t are the results of unpaired image to image translation network CycleGAN from S and T. Adaptation will be represent as $S_s \rightarrow T_s$. We use weakly supervised detection for fine-tuning the pre-trained adaptation network. Overall formulation is depicted in figure 1. This work is heavily inspired with Progressive Domain Adaptation [8], but we use two way generation process to bring the domain even more closer and Weakly supervised detection network for fine-tuning the adaptation network using pseudo labels [9].

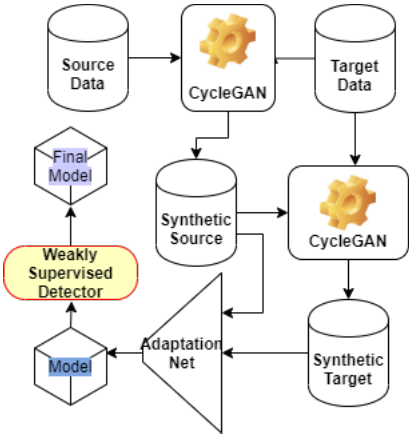

Figure 1: Proposed architecture: we create synthetic source S_s by translating source to the target using CycleGAN. Similarly we create synthetic target T_s by translating target to synthetic source S_s.S_s and T_s are fed to adaptation network.Trained model is fine-tuned using weakly supervised detector.

3.2 Synthetic image Generation

We used CycleGAN framework for unpaired image to image generation.In first phase of generation we translate source S to target T. Synthetic source S_s will have source domain content and target domain style. Now the newly generated source will be our synthetic source S_s. In our second generation we translate target$T$ toward synthetic source S_s and outcome will be denoted as synthetic target T_s. It uses adversarial loss [7] as in equation (1). G tries to generate image $G(x)$ which should be similar to images of domain D. Additionally Cycle consistency loss [14] (equation 2) is used to map the generated image back to their source. It helps to create natural looking images and avoid mode collapse. Function F maps the generated image back to the original image by minimizing the

L1 loss between reconstructed and original image.

$$L_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y\sim pdata(y)}[\log(D_Y(y))] + \mathbb{E}_{x\sim pdata(x)}[\log(1 - D_Y(G(x))] \quad (1)$$

$$L_{cyc}(G, F) = \mathbb{E}_{x\sim pdata(x)}[\| F(G(x)) - x \|_1] + \mathbb{E}_{y\sim pdata(y)}[\| G(F(y)) - y \|_1] \quad (2)$$

Overall loss will be as shown in equation (3) where λ is a regularization parameter.

$$L(G, F, D_x, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_x, Y, Y) + \lambda * L_{cyc}(G, F) \quad (3)$$

Optimal G and F will be achieved by solving equation (4)

$$G^*, F^* = arg \min_{G,F} \max_{D_x, D_y} L(G, F, D_x, D_Y) \quad (4)$$

3.3 Detection network

We have used Faster-RCNN [18] for the object detection task. It uses region proposal network**(RPN)** along with a network for classification and localization. Training happens in interleaved fashion, where we train **(RPN)** first and generate regions proposals. We freeze RPN and fed the region proposals to classification and localization network for training. There will additional RPN loss along with regular classification and regression loss as in equation (5).

$$L_{det} = L_{rpn_{cls}} + L_{rpn_{reg}} + L_{cls} + L_{reg} \quad (5)$$

3.4 Adaptation Step

Adaptation network consists of encoder network($\mathbb{E}_{en}$) discriminator network($\mathbb{D}_{dis}$) and detection network ($\mathbb{D}_{det}$).Training of adaptation network begins after having synthetic source S_s and synthetic target T_s as shown in proposed architecture in figure 1. We try to adapt from synthetic source domain for which we have the annotations to the synthetic target domain for which we don't have the annotations.Training of encoder network $\mathbb{E}_{en}$ and discriminator network $\mathbb{D}_{dis}$ happens similar to the generative adversarial network training.In the end Encoder network $\mathbb{E}_{en}$ will give domain invariant feature representation of the input image. Detection network $\mathbb{D}_{det}$ will update the weights based on the domain invariant inputs from the $\mathbb{E}_{en}$. It adjust the weight such that target domain images will get detected correctly. Similarly $\mathbb{E}_{en}$ keep updating the network weights such that generated feature will be domain invariant and closer to synthetic source S_s. Negative gradient flows from discriminator $\mathbb{D}_{dis}$ to encoder $\mathbb{E}_{en}$ during back-propagation as suggested in [6]. Adversarial loss is used for training the $\mathbb{E}_{en}$ and binary cross entropy loss for training the $\mathbb{D}_{dis}$ network. Generated images are having wide range and some of them act as an outlier. It affects the model performance. To counter it we use a weighting scheme with the help of trained discriminator network of CycleGAN as suggested in [8]. More weights re given to the samples which are closer to the source domain and it is formulates in equation (6) where $p_T(\mathbf{I})$ is probability of image ($\mathbf{I}$) belonging to target domain. Similarly, $p_S(\mathbf{I})$ is probability of image ($\mathbf{I}$) belonging to source.

$$D_{cycle}[\mathbf{I}] = \frac{p_T(\mathbf{I})}{(p_T(\mathbf{I}) + p_S(\mathbf{I}))} \quad (6)$$

Overall min-max loss is shown in equation (7) which consist of weighted detection loss for the samples belonging to synthetic source S_s and discriminator loss for all the samples. L_{dis} in equation (7) is cross entropy loss

$$E^*_{en}, D^*_{dis} = \min_{E_{en}} \max_{D_{dis}} D_{cycle}(I_{S_s}) L_{det}(I_{S_s}) + \lambda_{disc}[L_{dis}E_{en}(I_{S_s}) + L_{dis}E_{en}(I_{T_s})] \quad (7)$$

3.5 Active Adversarial Domain Adaptation

We wanted to explore the effectiveness of proposed method for Active learning. Once we have the trained adaptation network, we calculate the score based on the formulation given in equation (8) [20].

$$S(x) = \frac{1 - G_d(G_f(x))}{G_d(G_f(x))} \times H(G_y(G_f(x))) \quad (8)$$

Where $G_d(x) = \frac{p_s(x)}{p_s(x)+p_t(x)}$ and $H(G_y(G_f(x)))$ is entropy of score of the detection network $S(x)$ will be high for choose those samples which were hardest to detect from the detection network. I used the same architecture as in figure 1 except in each cycle we use oracle to get the annotation of b number of samples from the target dataset. Annotated target dataset is added to the synthetic source domain S_s and we retrain the adaptation network with $(L_s \cup L_t, U_t - L_t)$ dataset. L_t are the number of annotated target samples added to source and removed from unlabelled target. In each cycle of training we keep adding some samples from the target to the source and training continues. Modified architecture is shown in figure 2

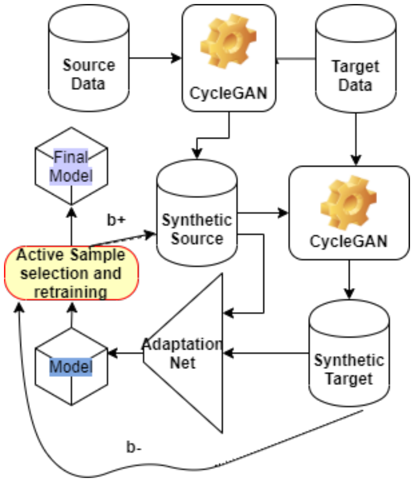

Figure 2: Active Learning Proposed architecture: we create synthetic source S_s by translating source to the target using CycleGAN. Similarly we create synthetic target T_s by translating target to synthetic source S_s.S_s and T_s are fed to adaptation network.In each cycle we get the annotation of b samples from the oracle and add it to S_s and remove those sampels from T_s.

3.6 Weakly Supervised Detection

Main difference between source and target domain lie in low level features such as color and texture. Domain gap between generated synthetic source and target distribution (S_s, T_s) are closer compared to natural source and target distribution (S, T). Adaptation step helps to learn domain invariant features. Weakly supervised detection help us to make detection more robust by fine-tuning the weights with he help of generated pseudo labels. We have reduced the learning rate for fine-tuning step by half.

4 Experiments

4.1 Datasets

KITTI: KITTI dataset [4] is taken while driving in cities, rural and highways. There are 7,481 images in the training set and it is used as a source domain for Cross camera Adaptation experiment.

CityScape: CityScape dataset [2] is collection of images with city street scenarios. It was taken while driving and camera was mounted with the car.It contains 2975 training images and 500 validation images

Foggy CityScape Foggy CityScape dataset [19] is build on CityScape dataset [2].
Dataset simulates the foggy weather condition with 3 levels of fog with the help of depth map information from the CityScape dataset. SO the dataset size for Foggy CityScape is three times the dataset size of CityScape.

Table 1: KITTI to CityScape Adaptation without fine-tuning

Training Method	Average Precision
Faster R-CNN	28.8
Progressive Domain Adaptation	43.9
Ours (Adaptation without Synthetic)	38.20
Ours (Adaptation with synthetic without weighting)	40.38
Ours (Adaptation with Synthetic and weighting)	**43.8**
Progressive Domain Adaptation [8]	**43.9**
Oracle	**55.80**

4.2 Image generation

Image generation happened in two phases. For cross camera adaptation, KITTI dataset is our source and CityScape is our target.In 1st phase we translated KITTI to CityScape resulting in synthetic KITTI. In 2nd phase, we translate CityScape dataset to synthetic KITTI resulting in synthetic CityScape. Sample generation results are shown in figure 3. Similarly for cross weather adaptation, we start with CityScape and Foggy CityScape and it results in synthetic CityScape and synthetic Foggy CityScape. In order to fit into memory we downsample images while training and testing. We upsample with the same amount while training the adaptation network.

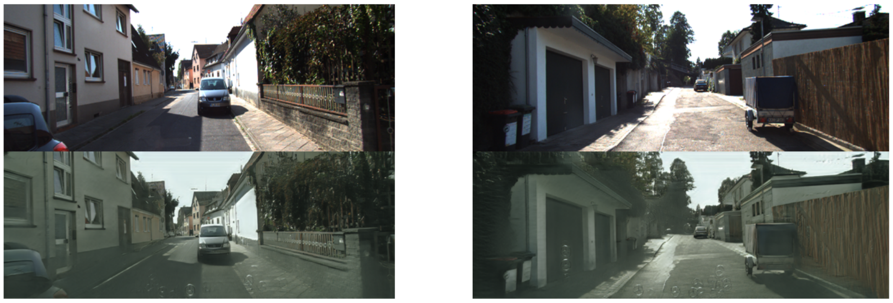

Figure 3: Generated Sample from KITTI to CityScape

4.3 Visualization

I extracted deep features from the domain kitti, cityScape, synthetic kitti and synthetic cityScape. I did the t-SNE plot for the qualitaive analysis. Data distribution became somewhat similar in synthetic image space compared to natura image space as shown in figure 5 and figure 4 respectively.

4.4 Cross Camera Adaptation

Our goal is to minimize the domain gap between KITTI and CityScape dataset in this experiment. Here domain gap is due to the different camera mounted on the vehicle along with different viewpoint and scene.Car class is available in both the classes. We report our average precision result on the validation set of CityScape dataset fro the car class.We used VGG16 architecture for encoder along with Faster-RCNN and discriminator.We have achieved almost similar AP using single step adaptation compared to two step adaptation for the Progressive domain adaptation [8]. Average precision scores are summarized in table 1. IoU is used to decide whether detected object object is true positive or true negative. IoU was set at 0.7.

4.5 Cross Weather Adaptation

We try to adapt weather condition in this experiment. It is a practical use-case where we need to adapt to different weather condition. Synthetic CityScape is our source domain and synthetic Foggy

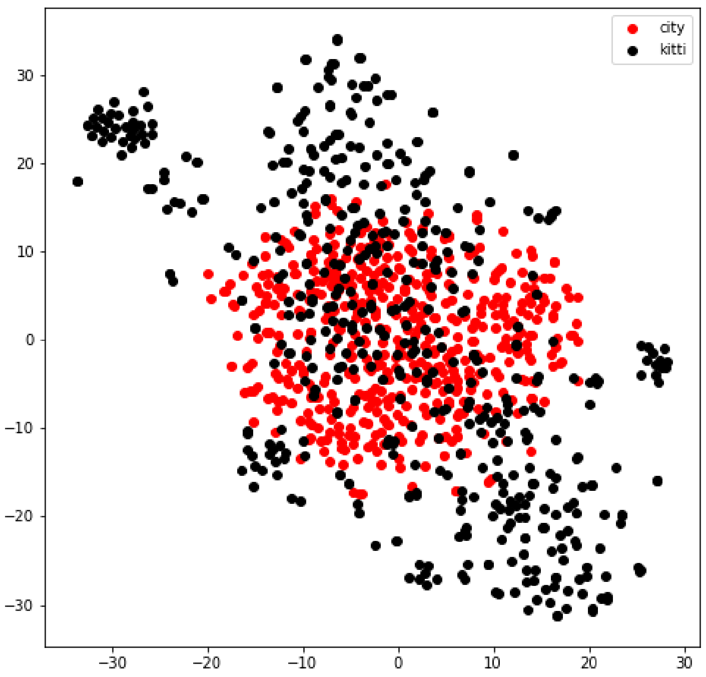

Figure 4: t-SNE plot for Natural Image space

Table 2: CityScape to Foggy CityScape Adaptation without fine-tuning

Class	Method			
Class name	Faster-RCNN	Progressive DA	ours(No Syn)	Ours (with Syn)
person	23.3	36.0	25.36	33.81
rider	29.4	45.5	32.44	43.73
car	36.9	54.4	38.34	53.42
truck	7.1	24.3	11.54	19.48
bus	17.9	44.1	23.53	43.33
train	2.4	25.8	3.4	16.64
motorcyle	13.9	29.1	17.1	26.03
bicycle	25.7	35.9	27.71	33.08
mAP	19.57	36.88	22.42	33.7

CityScape is target domain. We performed experiment for 8 classes.Average precision(AP) per class and mean average precision(mAP) is reported in table 2.

4.6 Active Adversarial Adaptation

I performed active adversarial domain adaptation experiment for cross camera adaptation where i am performing adaptation from KITTI to CityScape dataset. Experiment results are shown in table 3. Though the results are not very interesting as compared to the performance shown in [20] , but it can help us to extend it further for different task including classification because generation technique is not restricted.

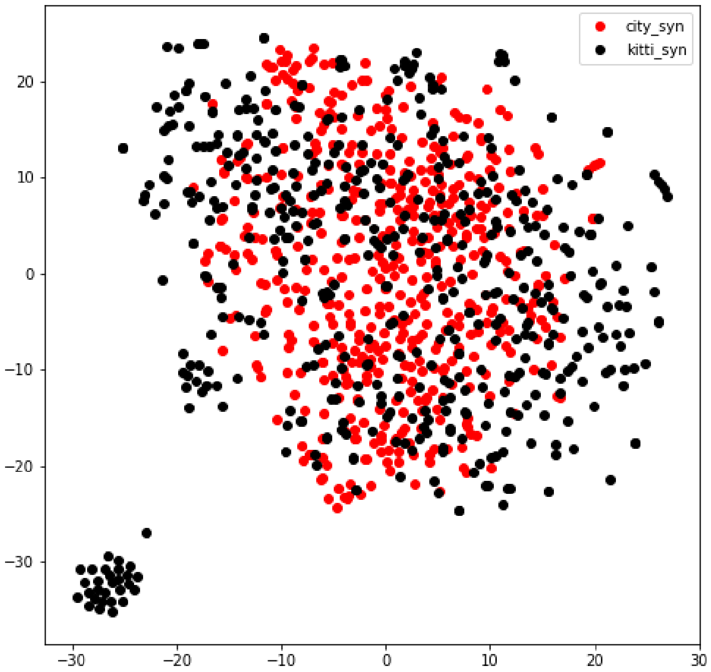

Figure 5: t-SNE plot for Synthetic image space

Table 3: KITTI to CityScape Active Adversarial Domain Adaptation

No. of target sample used per cycle	Average Precision
0	43.80
125	42.50
510	45.57
521	**46.13**

4.7 Weakly Supervised Detection

Weakly supervised detection will be used for fine-tuning of adaptation network. There are challenges of selecting hard samples which can also be detected very confidently with the pre-trained network. It helps to tune the trained weights and bring then very closer to the target domain so that they can be a good representation for the target domain. It is still an on-going task. Sample output from Pre-trained adaptation network are shown in figure 6

5 Conclusion

We explored image generation technique to crate a synthetic domain with lesser domain gap. It improves the performance on baseline and almost bring the performance closer to state of the art. I am experimenting with weakly supervised detection on top of existing trained model which will bring the robustness to our model.I also explored the usefulness of the proposed architecture for active adversarial adaptation.

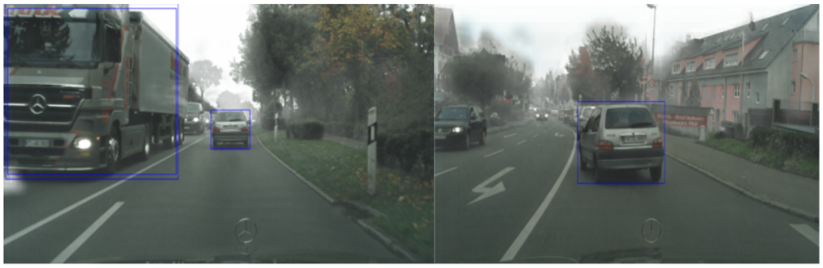

Figure 6: Adaptation network output for target domain sample

References

[1] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[3] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[5] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[6] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[8] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 749–757, 2020.

[9] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[11] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *European Conference on Computer Vision*, pages 350–365. Springer, 2016.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[14] A Radford, L Metz, and S Chintala. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2016.

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[16] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[17] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[19] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

[20] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 739–748, 2020.

[21] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2843–2851, 2017.

[22] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.