Domain Adaptation for Vehicle Detection from Bird's Eye View LiDAR Point Cloud Data

Khaled Saleh, Ahmed Abobakr, Mohammed Attia, Julie Iskander,
Darius Nahavandi, Mohammed Hossny, and Saeid Nahavandi

Deakin University, Australia

Abstract

Point cloud data from 3D LiDAR sensors are one of the most crucial sensor modalities for versatile safety-critical applications such as self-driving vehicles. Since the annotations of point cloud data is an expensive and time-consuming process, therefore recently the utilisation of simulated environments and 3D LiDAR sensors for this task started to get some popularity. However, the generated synthetic point cloud data are still missing the artefacts usually exist in point cloud data from real 3D LiDAR sensors. Thus, in this work, we are proposing a domain adaptation framework for bridging this gap between synthetic and real point cloud data. Our proposed framework is based on the deep cycle-consistent generative adversarial networks (CycleGAN) architecture. We have evaluated the performance of our proposed framework on the task of vehicle detection from a bird's eye view (BEV) point cloud images coming from real 3D LiDAR sensors. The framework has shown competitive results with an improvement of more than 7% in average precision score over other baseline approaches when tested on real BEV point cloud images.

1. Introduction

Recently, deep learning-based techniques such as convolution neural networks (ConvNets) have been achieving state-of-the-art results in many computer vision tasks such: object identification [18], scene understanding [24, 19], and human action recognition [21, 1, 23]. However, these techniques require a handful amount of labelled data for training them which is both time-consuming and cumbersome process to get for many tasks. Thus, the utilisation of synthetic data for training such techniques got some momentum over the past few years [20, 22]. With synthetic data, the process for obtaining ground-truth labels becomes much easier and automated most of the time. However, still, the utilisation of synthetic data is not entirely reliable because of its limitations when it comes to the generalisation to real data. In safety-critical applications such as a self-driving vehicle, one of the main sensors that are currently crucial for its development is the 3D LiDAR (Light Detection And Ranging) sensor. 3D LiDAR sensors can reliably provide 360° point cloud in traffic environment with coverage distance up to 200 meters ahead across different weather and lighting conditions. Thus, a number of deep-learning based techniques have recently been utilising its point cloud for many perception tasks for self-driving vehicles [29, 28]. One of the main reasons that the number of deep-learning techniques that rely on point-cloud data is not as much as the ones rely on visual data is the scarcity of labelled point cloud data. The labelling procedure for point cloud data is more complicated than visual data especially for tasks such as 3D object detection and per-point semantic segmentation. Thus, the usage of synthetic data has been explored, similar to

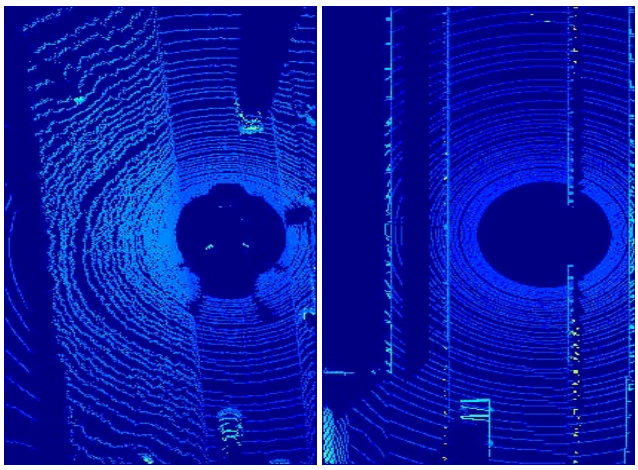

Figure 1. Sample of BEV images of real point cloud data (left) from a real Velodyne 3d LiDAR from KITTI dataset [9] and a synthetic point cloud data (right) from a simulated 3D LiDAR sensor from MDLS dataset [29].

the visual data modality data [20, 29]. However, the generalisation to real-point cloud data was rather limited due to the perfectness of the synthetic point cloud data (shown in Fig. 1, right) which is missing the artefacts usually exist in point cloud data from real 3D LiDAR sensors (shown in Fig. 1, left). These artefacts are such as the variability of the LiDAR beams intensities or the motion distortion as a result of the motion of the 3D LiDAR. Domain adaptation (DA) is one of the machine learning (ML) techniques that have been recently explored to bridge the aforementioned gaps between synthetic and real data domains [27]. In DA, the goal is to learn from one data distribution (referred to as the source domain) a perfect model on a different data distribution (referred to as the target domain). In traffic environments, DA has recently shown promising results for image translation between different domain pairs such as night/day, synthetic/real images and RGB/thermal images [31]. Since most of the previous DA techniques are based on 2D deep ConvNet architectures, thus their application on 3D point cloud data from 3D LiDAR sensors is not a straight forward task. On the other hand, the recent deep-learning based techniques that have been applied on perception tasks using 3D point cloud data, they managed to find a way to adopt the same 2D ConvNet architectures to work on the 3D point cloud data. One of the most common techniques was to project a top-down bird's eye view (BEV) of the point cloud data on a 2D plane (ie. ground). The representation of the 3D LiDAR point cloud data as a BEV was shown to be effective in many perception tasks for self-driving vehicles such as 3D object detection [15], road detection [4] and per-point semantic segmentation [6].

To this end, in this work, we will be proposing a DA approach for vehicle detection in real point cloud data from 3D LiDAR sensors represented as BEV images. The proposed DA approach will be a deep learning-based approach based on deep generative adversarial networks (GANs) [31]. For the vehicle detection task, it will be based on state-of-the-art deep object detection architecture YOLOv3 [18]. The rest of the paper is organised as follows. In Section 2, a brief introduction to the different DA approaches with emphasis on deep learning based approaches will be reviewed in addition to a quick review on GANs. Section 3, the methodology we followed for our proposed DA approach will be discussed thoroughly. Experiments and results are discussed in Section 4. Finally, Section 5 concludes.

2. Related Work

Commonly, there are two ways to achieve DA either by directly translating one domain to the other or by obtaining a common-ground intermediate pseudo-domain between the two domains. In the following, firstly a quick review of the work related to the DA approaches will be provided specifically the approaches based on the direct translation between domains. Then, a brief summary of the DA work between simulated and real domains done in the context of traffic environments will be discussed.

2.1. Adversarial Domain Adaptation

Historically, most of the work done on DA has been relying on the transformation between source and target domains based on linear representations [3, 10]. Until the emergence of the recent set of techniques based on non-linear transformation representations via neural networks [8, 26], which have achieved state-of-the-art results in a number of DA benchmarks [17, 14]. One of the most commonly non-linear-based representations DA approaches is the adversarial domain adaptation (ADA) approach [8]. ADA was inspired by the work done by Goodfellow et al. [11] on generative adversarial networks (GANs). In GANs, there are two deep neural networks trained simultaneously, namely a "generator" network and a "discriminator" network. The generator network, as the name implies, it generates new data instances using a uniform distribution, on the other hand, the discriminator network tries to decide whether or not this newly generated data instance has the same distribution as the training dataset distribution. Similarly, in ADA, it has the same two networks, this architecture is often referred to in the literature as the "conditional GAN". One of the most recently successful ADA architectures is the Cycle-Consistent GAN (CycleGAN) [31] architecture. In CycleGAN, it is essentially comprised of two conditional GAN networks. The first network works on the transformation from the source domain (S) to the target domain (T), $S \rightarrow T$, while the other one works on the transformation in the opposite direction, $T \rightarrow S$. The additional contribution for CycleGAN architecture was the introduction of a new loss function they call it the cycle-consistency loss function. This new loss function assures that if the two conditional GANs networks are connected, they will produce the following identity mapping: $S \rightarrow T \rightarrow S$.

2.2. DA Between Synthetic and Real for Perception Tasks

In the context of traffic environments, a number of perceptions tasks has been utilising the DA approach to bridge the gap between real domains from physical sensors and synthetic domains from simulated sensors [31, 2, 30, 25]. It is worth noting that all of these works were only exploring one type of sensors which was cameras either RGB (monocular/stereo) or thermal. For example, in [31], a number of DA between different domains were introduced based on the CycleGAN architecture. For instance, they addressed the semantic segmentation task between the day and night domains on unpaired visual images from multiple road-based datasets. Similarly, in [2, 5] the authors trained a ConvNet model on synthetic depth and RGB images from the

famous game GTA in order to estimate a synthetic monocular depth image and localise objects respectively. In the testing/inference phase, they took an input real RGB image from the KITTI dataset [16] and with the help of a CycleGAN architecture, they transformed the real RGB image into a synthetic GTA game like RGB image. Then, they passed the synthetic RGB image to their initial trained model to estimate a synthetic depth image. Eventually, they used the same CycleGAN network again to adopt the estimated depth image from the synthetic image domain to a real RGB image domain. On the other hand, in [30] Zhang et al. proposed deep-learning based approach for thermal infra-red object tracking. To overcome the scarcity of thermal images dataset, they utilised DA based on the CycleGAN architecture to transform images from visual domain to the thermal infra-red domain.

3. Proposed Methodology

The main focus of this work is to provide a framework for bridging the gap between real and synthetic point cloud data represented as BEV images for the vehicle detection task. That being said, the same framework can still be used for other perceptions tasks on point cloud data such as semantic segmentation or object tracking. In this section, we will first provide our formulation for the problem at hand. Then subsequently, we will break-down the building blocks of the proposed framework.

3.1. Problem Formulation

In our formulation for the vehicle detection task from real BEV point cloud data, we are proposing a framework consisting of two stages. In the first stage of our framework, we train a CycleGAN model between unpaired synthetic BEV point cloud data and real BEV point cloud data. The trained model, in returns, learns a transformation from synthetic BEV point cloud data to real BEV point cloud data and vice versa. As a result, given any annotated synthetic BEV point cloud dataset with vehicles, the trained CycleGAN model will transform that dataset to an annotated real-like BEV point cloud data. Finally, using the transformed dataset, we could train another ConvNet-based model for the vehicle detection task in real BEV point cloud data.

3.2. Deep Unsupervised DA via Cycle-Consistent GANs

In this work, we will be exploring the CycleGAN architecture for the task of DA between real BEV point cloud domain and synthetic BEV point cloud domain. One of the advantages of the CycleGAN architecture in the context of DA is it can learn transformation between source and target domains without any supervised one-to-one mapping between the two domains. This is beneficial for our task because it is almost impossible for us to have the same traffic scenario and environment captured in both real BEV point cloud data and synthetic BEV point cloud data. However, we can have a handful amount of BEV point cloud data from each domain separately that represent the distribution of that domain. More formally, given our two domains S, R of the synthetic and the real BEV point cloud data domains. Then, the objective of our adopted CycleGAN-based DA approach (shown in Fig. 4) is to map between the distributions $s \sim \mathbb{P}_d(s)$ and $r \sim \mathbb{P}_d(r)$ from the synthetic and the real BEV point cloud domains respectively. The proposed CycleGAN-based DA approach achieve this mapping via the two generators, $G_{S \to R}$ and $G_{R \to S}$ and the two discriminators D_S and D_R. The generator $G_{S \to R}$ will try to map the input source synthetic BEV point cloud image to some target real BEV point cloud image. While the generator $G_{R \to S}$ is trying to map the generated BEV point cloud image from the real target domain back to its original source domain. The discriminator D_S, on the other hand, is trying to differentiate between a BEV point cloud image $s \in S$ and a generated BEV point cloud image from $G_{R \to S}$. Conversely, the discriminator D_R will be trying to distinguish between a BEV point cloud image $r \in R$ and a generated BEV point cloud image from $G_{S \to R}$. The two generators networks are deep ConvNet models. The main building blocks of them are three blocks, namely the encoder, the transformer and the decoder respectively. The encoder's job is to extract features on multiple levels progressively by down-sampling them from the input BEV point cloud image from both domains. The transformer, on the other hand, takes the extracted features vector encoder in the source domain and transform it into another feature vector in the opposite target domain. The decoder finally up-sample the transformed features vector back to the original shape and dimensionality as it was before going through the encoder. The architecture we used for that combination of encoder, transformer and decoder of our generator networks is based on the architecture proposed in [13]. The encoder in this architecture consists of two convolution layers, while the transformer consists of nine ResNet blocks and the decoder consists of two de-convolution/transposed convolution layers. The two discriminators architecture is a deep ConvNet model as well. They are based on the PatchGAN architecture from [12], which consists of three consecutive convolution layers for feature extraction in patches and a final 1D-convolution layer for the decision whether its input BEV point cloud image is fake or not. In order to train the proposed CycleGAN-based DA approach for our task, we will be utilising the adversarial loss for the two generators that we have discussed above along with their corresponding discriminators. The first loss for the transformation from domain S to domain R is as follows:

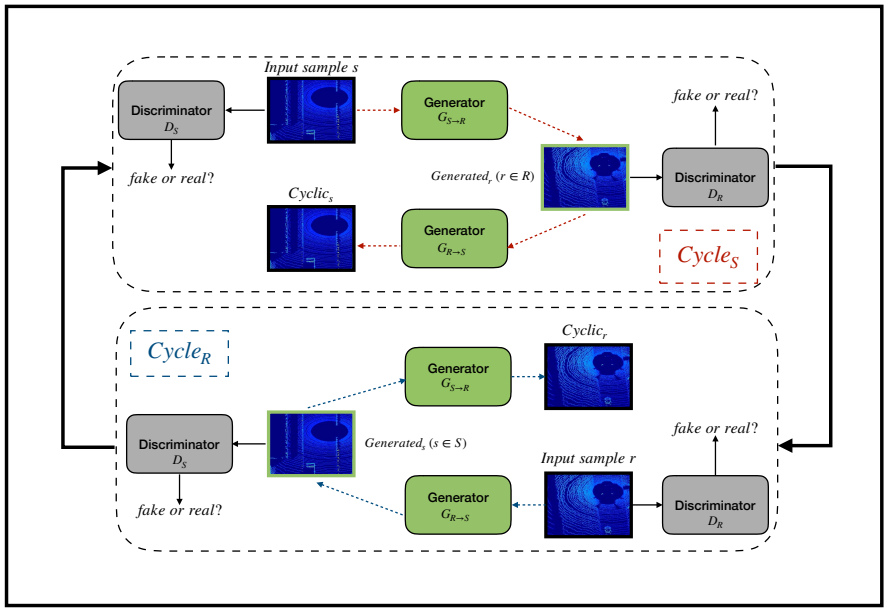

Figure 2. Proposed CycleGAN-based DA framework for the vehicle detection task in BEV point cloud images. The framework has two internal cycles, namely $Cycle_S$ and $Cycle_R$. In $Cycle_S$, the input sample s of synthetic BEV point cloud image goes firstly through the generator $G_{S\to R}$ which its output is interrogated by the discriminator D_R. The generated sample r is then goes through the other generator $G_{R\to S}$ for reconstructed the original input s sample. The same process goes for the second cycle $Cycle_S$.

$$\mathcal{L}_{adv_{S\to R}} = \min_{G_{S\to R}} \max_{D_R} \mathop{\mathbb{E}}_{r\sim\mathbb{P}_d(r)} [\log D_R(r)] + \mathop{\mathbb{E}}_{s\sim\mathbb{P}_d(s)} [\log(1 - D_R(G_{S\to R}(s)))] \quad (1)$$

where S is the synthetic BEV point cloud data domain and $\mathbb{P}_d(s)$ is its data distribution.

Similarly, the second loss for the transformation from domain R to domain S is as follows:

$$\mathcal{L}_{adv_{R\to S}} = \min_{G_{R\to S}} \max_{D_S} \mathop{\mathbb{E}}_{s\sim\mathbb{P}_d(s)} [\log D_S(s)] + \mathop{\mathbb{E}}_{r\sim\mathbb{P}_d(r)} [\log(1 - D_S(G_{R\to S}(r)))] \quad (2)$$

Additionally, in order to penalise the generators of the trained model to generate more realistic BEV point cloud data from each domain S and R, the following third loss is added.

$$\mathcal{L}_{cyc} = \|G_{R\to S}(G_{S\to R}(s)) - s\|_1 + \|G_{S\to R}(G_{R\to S}(r)) - r\|_1 \quad (3)$$

where $\mathcal{L}_{cyc}$ is the cycle-consistency loss which ensures the identity mapping of the each transformed sample BEV point cloud image back to its original source. Given the three losses from Eq. 1, 2, 3, the objective loss function for the proposed CycleGAN-based DA approach is as follows:

$$\mathcal{L} = \mathcal{L}_{adv_{S\to R}} + \mathcal{L}_{adv_{R\to S}} + \lambda\mathcal{L}_{cyc} \quad (4)$$

where λ is equal to 10 which was chosen empirically.

Finally, since the objective of training any deep ConvNet model is to minimise a certain loss function, which in our case is the joint loss function in Eq. 4. Thus, we will be using the Adam optimiser for minimising our objective joint loss function using a learning rate of 0.001.

3.3. Vehicle Detection in BEV Point Cloud Data via YOLOv3

For the vehicle detection task, we will be the adopting state-of-the-art single stage deep ConvNet architecture for object detection, You Only Look Once (YOLOv3) architecture. Internally, YOLOv3 relies on k-means clustering to have prior bounding boxes "anchors" of a potential region of interests (ROIs) in the input image which goes through a total of 53 convolution layers to extract features from them on 3 different scales. YOLOv3 in returns predicts the four coordinates for the bounding box, an objectness score for each bounding box, and class score for the object that the bounding box may contain. The four coordinates are predicted using a sigmoid function. The objectness score is predicted using a logistic regression which is set to 1 if the bounding box of one of the anchors overlaps with a ground truth bounding box. The class score of a bounding box is predicted via multinomial logistic classifiers which is better than the traditional soft-max classifier when it comes to multi-label classification task such as object detection. More specifically, in our vehicle detection task from BEV point cloud images, we relied on the YOLOv3-416 derivative architecture, which as the name implies works on input

images with a resolution of $416H \times 416W$.

4. Experiments

In this section, we will firstly discuss the datasets we have used for training and validating our trained models. Secondly, the performance of our models will be quantitatively and qualitatively evaluated.

4.1. Datasets

For the task of the DA between synthetic and real BEV point cloud images, we relied on two datasets. The first dataset is the recently released Motion-Distorted LiDAR Simulation (MDLS) dataset introduced in [29]. This dataset represents the synthetic domain S of our CycleGAN-based DA approach discussed in Section 3.2. The MLDS dataset was generated from high fidelity simulated urban traffic environments from the CARLA simulator [7] using a simulated Velodyne HDL-64E sensor. The dataset is originally meant for studying the effect of the motion distortion resulted from a moving vehicle-based 3D LIDAR sensor on the generated point cloud data. The dataset consists of two sequences of point cloud data from urban traffic environment involving between 60 to 90 moving vehicle, each one with an average duration of five minutes which results in total 6K point cloud scans. The dataset was annotated with the position of the vehicles in the scene. For our DA task, we first preprocessed the point cloud scans in order to get a BEV image of each scan according to the method introduced in [15]. As a result, we get a total of 6K BEV point cloud images similar to the right image shown in Fig. 1. The second dataset we utilised for the real domain R of our CycleGAN-based DA approach is the BEV benchmark data from the KITTI dataset [9]. The BEV benchmark data consists of 7481 training images and point cloud scans and 7518 test images and point cloud scans. The point cloud data was captured using a real 3D LiDAR sensor the Velodyne HDL-64E sensor. The dataset contains annotations for multiple objects in the traffic scene such as vehicles, pedestrians and cyclists. Similar to the pre-processing step we have done for the MLDS dataset we did it as well for the KITTI dataset in order to get BEV point cloud images like the one shown on the left in Fig. 1. In our experiments for training our CycleGAN-based DA approach, we used a total 6K BEV point cloud images from the MLDS dataset and the 7481 BEV point cloud images of the training split from the KITTI dataset. Similarly, for the task of the vehicle detection from BEV point cloud images we used the same aforementioned two datasets (MLDS and KITTI) in addition to the domain adapted BEV images from synthetic to real for training our YOLOv3 model. Since our ultimate goal in the vehicle detection task is to identify vehicles in real BEV point cloud images. Thus, we further split the total 7481 real BEV images from the KITTI dataset into 4K for training our YOLOv3 model and 3481 for testing the model.

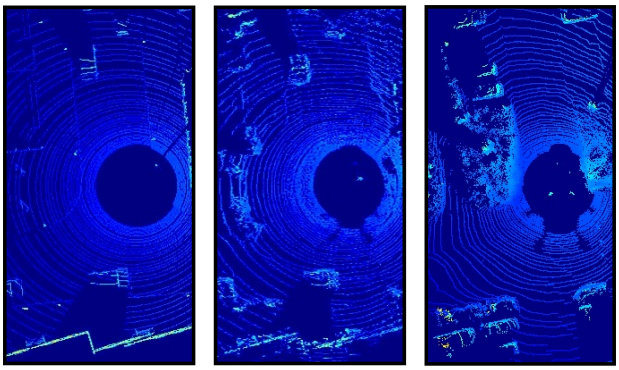

Figure 3. Qualitative results for the proposed CycleGAN-based method for DA between synthetic and real BEV point cloud data. a) the input synthetic BEV point cloud image from [29], b) the transformed real BEV point cloud image using the proposed method and c) the correlated real BEV point cloud image from the KITTI dataset [9].

4.2. Results and Discussion

Firstly, in order to evaluate the effectiveness of our proposed CycleGAN based DA approach for the vehicle detection task from real BEV point cloud images. In fig. 3, we show qualitative results of the trained CycleGAN-based DA approach between synthetic and real BEV point cloud images. In the first row of the figure is the input synthetic BEV point cloud image to our model. The second row represents the output from the generator $G_{S \rightarrow R}$ of our trained CycleGAN model. The third row shows one sample of a real BEV point cloud image from the KITTI dataset. As it can be noticed, the generated BEV point cloud from our CycleGAN model is mimicking and trying to be consistent with the same structure exist in the real BEV point cloud image from KITTI. More specifically, the generated image captures pretty well the structure of the vehicles and the distortion/noise artefacts from resulting from the real Velodyne 3D LiDAR sensor. For having more quantitative evaluation of our proposed CycleGAN based DA approach for the vehicle detection task, we trained two YOLOv3 models, the first one $YOLO_S$ is trained using the 6K synthetic BEV point cloud images, while the other one $YOLO_R$ is trained using the same 6K BEV point cloud images but the DA versions of them after feeding them to our trained CycleGAN model and getting its predicted DA real BEV point cloud images. Furthermore, we trained three additional YOLOv3 models with the only difference in the type of training data. The first model $YOLO_K$ which as the name implies is trained on the 4K training split BEV point cloud images from the KITTI dataset. The second model $YOLO_{KS}$ is

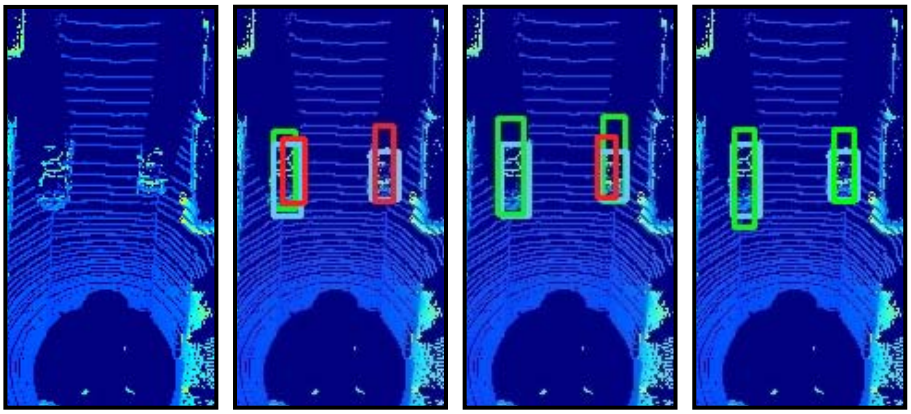

Figure 4. Qualitative results on the KITTI BEV point cloud dataset for the vehicle detection task. From left to right, a) the input BEV image , b) bounding box detections from $YOLO_K$ model, c) bounding box detections from $YOLO_{KS}$ model, d) bounding box detections from $YOLO_{KR}$ model.

Table 1. Comparison between our 5 trained YOLOv3 models on the same testing split BEV point cloud images from the KITTI dataset [9]. Higher is better.

Model	Training Data	Average Precision (AP)%
$YOLO_S$	SYN (only)	29.93
$YOLO_R$	DA (only)	**34.78**
$YOLO_K$	KITTI (only)	57.26
$YOLO_{KS}$	KITTI+SYN	59.16
$YOLO_{KR}$	KITTI+DA	**64.29**

trained using on the 4K images from the KITTI dataset with an additional 6K synthetic BEV point cloud image from the MLDS dataset. The third and final model $YOLO_{KR}$ is trained using the same amount of data to the $YOLO_{KS}$ model, however instead of the MLDS synthetic BEV images we used the DA version predicted from our CycleGAN model.

In Table 1, we report the performance of the total 5 YOLOv3 models we mentioned earlier when all are tested on the same 3481 testing real BEV point cloud images from the KITTI dataset. The evaluation metric we used is the average precision score (AP) which summarises the precision-recall curve that commonly used for evaluating object detectors. As it can be noticed from the table, the $YOLO_R$ model outperformed the $YOLO_S$ with more than 4% in AP score which proves our claim that our CycleGAN-based DA approach for the BEV point cloud images are more efficient than pure synthetic ones for the vehicle detection task. Additionally, the best performing model with 64.29% in AP score is the $YOLO_{KR}$, which again proves the benefits of using domain adapted BEV point cloud images over the purse synthetic ones. This prevalent from Table 1 by the low AP scores from the $YOLO_K$ and the $YOLO_{KS}$ models which achieved only AP score of 57.26% and 59.16% respectively. For a qualitative measuring of the performance of the trained YOLOv3 models, in Fig. 4, we show a) input sample BEV point cloud image, b), c) and d) the detected bounding boxes (in green colour) from models $YOLO_K$, $YOLO_{KS}$ and $YOLO_{KR}$ respectively. The ground truth annotations are highlighted in the light blue colour, while the false or miss-detected objects are highlighted in red colour. As it can be shown, our model $YOLO_{KR}$ gives an accurate detection with the lowest false-positive rate.

5. Conclusion

In this work, we have introduced a framework for domain adaptation between synthetic and real BEV point cloud images for the vehicle detection task. The proposed framework utilises deep generative adversarial networks, CycleGAN for the domain adaptation task. Then, given the domain adapted BEV point cloud images we trained a series of object detection models based on state-of-the-art deep ConvNet-based model, YOLOv3. The trained models have shown the effectiveness of the proposed DA approach for the vehicle detection task from real BEV point cloud images. Furthermore, we have evaluated the performance of the trained models on the testing split from real BEV point cloud images from the KITTI dataset. The best performing model was the one utilising our domain-adapted BEV point cloud images which achieved the highest average precision score of 64.29% with an improvement of more than 7% over the compared baseline approaches.

References

[1] A. Abobakr, M. Hossny, H. Abdelkader, and S. Nahavandi. Rgb-d fall detection via deep residual convolutional lstm net-

works. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–7, Dec 2018. 1

[2] A. Atapour-Abarghouei and T. P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2810, 2018. 2

[3] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics, 2006. 2

[4] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde. Fast lidar-based road detection using fully convolutional neural networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1019–1024. IEEE, 2017. 2

[5] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 2

[6] A. Dewan, G. L. Oliveira, and W. Burgard. Deep semantic classification for 3d lidar data. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3544–3549. IEEE, 2017. 2

[7] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 5

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 5, 6

[10] P. Germain, A. Habrard, F. Laviolette, and E. Morvant. A pac-bayesian approach for domain adaptation with specialization to linear classifiers. In *International conference on machine learning*, pages 738–746, 2013. 2

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. 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. 3

[13] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 3

[14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[15] B. Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE, 2017. 2, 5

[16] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 3

[17] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 2

[18] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018. 1, 2

[19] K. Saleh, M. Attia, M. Hossny, S. Hanoun, S. Salaken, and S. Nahavandi. Local motion planning for ground mobile robots via deep imitation learning. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4077–4082. IEEE, 2018. 1

[20] K. Saleh, M. Hossny, A. Hossny, and S. Nahavandi. Cyclist detection in lidar scans using faster r-cnn and synthetic depth images. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017. 1, 2

[21] K. Saleh, M. Hossny, and S. Nahavandi. Cyclist trajectory prediction using bidirectional recurrent neural networks. In *Australasian Joint Conference on Artificial Intelligence*, pages 284–295. Springer, 2018. 1

[22] K. Saleh, M. Hossny, and S. Nahavandi. Effective vehicle-based kangaroo detection for collision warning systems using region-based convolutional networks. *Sensors*, 18(6):1913, 2018. 1

[23] K. Saleh, M. Hossny, and S. Nahavandi. Long-term recurrent predictive model for intent prediction of pedestrians via inverse reinforcement learning. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2018. 1

[24] K. Saleh, R. A. Zeineldin, M. Hossny, S. Nahavandi, and N. El-Fishawy. End-to-end indoor navigation assistance for the visually impaired using monocular camera. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3504–3510. IEEE, 2018. 1

[25] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017. 2

[26] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017. 2

[27] M. Wang and W. Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2

[28] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. *arXiv preprint arXiv:1809.08495*, 2018. 1

[29] D. J. Yoon, T. Y. Tang, and T. D. Barfoot. Mapless online detection of dynamic objects in 3d lidar. *arXiv preprint arXiv:1809.06972*, 2018. 1, 2, 5

[30] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan. Synthetic data generation for end-to-end thermal infrared tracking. *IEEE Transactions on Image Processing*, 28(4):1837–1850, 2019. 2, 3

[31] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 2