An Adversarial Training based Framework for Depth Domain Adaptation

Jigyasa Singh Katrolia[1], Lars Krämer[2], Jason Rambach[1], Bruno Mirbach[1] and Didier Stricker[1,2]
[1]*German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany*
[2]*Technische Universität Kaiserslautern, Kaiserslautern, Germany*
{*jigyasa_singh.katrolia, jason_raphael.rambach, bruno_walter.mirbach, didier.stricker*}*@dfki.de,*
l_kraemer14@informatik.uni-kl.de

Keywords: Domain adaptation, adversarial training, time-of-flight, synthetic data, depth image, image translation

Abstract: In absence of sufficient labeled training data, it is common practice to resort to synthetic data with readily available annotations. However, some performance gap still exists between deep learning models trained on synthetic versus on real data. Using adversarial training based generative models, it is possible to translate images from synthetic to real domain and train on them easily generalizable models for real-world datasets, but the efficiency of this method is limited in the presence of large domain shifts such as between synthetic and real depth images characterized by depth sensor and scene dependent artifacts in the image. In this paper, we present an adversarial training based framework for adapting depth images from synthetic to real domain. We use a cyclic loss together with an adversarial loss to bring the two domains of synthetic and real depth images closer by translating synthetic images to real domain, and demonstrate the usefulness of synthetic images modified this way for training deep neural networks that can perform well on real images. We demonstrate our method for the application of person detection and segmentation in real-depth images captured in a car for in-cabin person monitoring. We also show through experiments the effect of using target domain image sets captured using different types of depth sensors on this domain adaptation approach.

1 INTRODUCTION

Depth information on its own or in addition to other sensory information can be leveraged by computer vision algorithms to gain a more complete understanding of the real-world. Deep learning based computer vision algorithms however, require plenty of annotated data to learn a generalizable mapping from input features to output labels. Acquiring such a large annotated dataset is a tedious task and demands abundant human effort and time. In such scenarios, synthetic images can be used as a reliable replacement for real-world data due to their ease of acquisition and ready availability of ground truth annotations. Even then, deep learning networks trained on synthetic data do not transfer well to real data; a phenomenon referred to as domain shift.

A solution to cope with the domain shift is to transform the synthetic images themselves to look like real ones and use them in conjunction with the original annotations corresponding to the synthetic images to train deep neural networks for the target task. This domain adaptation framework which relies on image translation networks has been explored to some degree (Dundar et al., 2018; Mueller et al., 2018; Shrivastava et al., 2017) since the introduction of adversarial training based image generation models called Generative Adversarial Networks or GANs (Goodfellow et al., 2014). So far this approach has been used mostly to translate synthetic RGB to real RGB images, for example from GTA (Richter et al., 2016) to Cityscapes (Cordts et al., 2016), and to a much lesser extent to translate from synthetic depth to real depth image domain. In the occasional cases

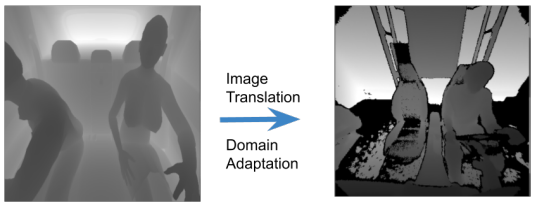


Figure 1: In our image translation based domain adaptation framework we use synthetic depth images (left) and real depth images (right) of in-car cabin scenes as source and target domain datasets respectively.

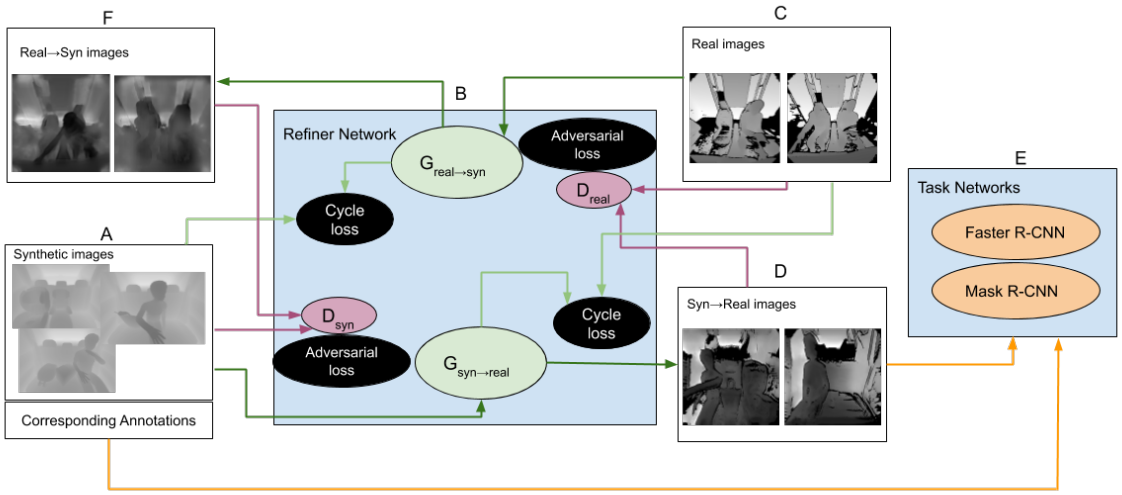


Figure 2: Overview of our image translation based domain adaptation approach. The Refiner network comprises a couple of generator-discriminator pairs each tasked with learning the representation of each of the source and target image domains. The generator $G_{syn \rightarrow real}$ takes images from synthetic domain A and maps them to real domain D (dark green). The discriminator D_{real} gets both the transformed images D and real images C to calculate the adversarial loss (pink). A cycle loss is computed to minimize difference between a synthetic image and its output after it has gone through the both generators (light green). A similar cycle exists for real images as well. Finally, two task networks E are trained on realistically refined synthetic images D (orange).

where it has been done, this translation is limited to scenes containing only a single foreground object (Shrivastava et al., 2017; Mueller et al., 2018). This problem is challenging for depth images as the gap between synthetic and real domains is not just in the lack of "realism" but also in the specific noise patterns which exist in real images due to properties of both the scene as well as the depth sensor.

In this paper we address unsupervised image translation based domain adaptation from the synthetic depth to real depth image domain and demonstrate its usability for in-car person detection and person segmentation tasks. With the advent of autonomous and driver-less vehicles, it is imperative to monitor the entire in-cabin scene in order to realize active and passive safety functions, as well as advanced human-vehicle interfaces, to increase the acceptance of such vehicles by the masses. Our main contribution thus lies in an image translation based domain adaptation method that can realistically refine synthetic depth images to reproduce the noise patterns typical for a depth sensor like missing pixels along object edges and depth holes as shown in Figure 1. Our secondary contribution is to show that using these refined images, one can train deep neural networks that perform considerably better than networks trained on only synthetic depth images, and with fine-tuning we can even surpass the performance of networks trained on real data. To the best of our knowledge, we are the first to use this approach for domain adaptation from synthetic depth images to real depth images of complete scenes. Our final contribution is to provide a quantitative and qualitative comparison of using different depth sensors, that work on both time-of-flight and pattern projection principles, to capture target domain image sets for domain adaptation. Altogether, our experiments show that domain adaptation coupled with adversarial training based image translation can be used to extract greater profit out of synthetic data.

2 RELATED WORK

2.1 Adversarial Training Based Image Translation

The seminal work on Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) presented a method to learn a generative model which can map a random noise vector to image samples that look like they are taken from a target probability distribution. GANs rely on an adversarial loss to learn a generative model which can generate fake images reasonably identical to real images in the presence of a dis-

criminator which tries to discern if the output of the generator is taken from the target probability distribution or from the generator. The authors of (Isola et al., 2017; Karacan et al., 2016) performed supervised image translation where the output translated image is conditioned on an input image. This requires paired input images from source and target domain. In contrast to these, (Zhu et al., 2017; Liu et al., 2017) performed unsupervised image translation where they do not require any supervision at 'pair' level, reducing the effort required to collect paired training datasets.

2.2 Image Translation Based Domain Adaptation

Most of the early works related to deep learning based visual domain adaptation focus on feature level adaptation between source and target domains. They achieve this by either mapping source and target distribution to a common feature space (Hoffman et al., 2017; Hu et al., 2015; Motiian et al., 2017) or by learning domain invariant features which can be extracted at inference time to make predictions (Ghifary et al., 2014; Tzeng et al., 2014). An alternative approach is to perform domain adaptation in the image space by translating images from source domain to target domain and then using these translated images for training the final task network along with the source annotations. For synthetic to real domain adaptation (Dundar et al., 2018) used a style transfer network based on FastPhotoStyle (Li et al., 2018) to stylize synthetic images to look realistic using randomly paired input images. Building upon CycleGAN, many works like CyCADA (Hoffman et al., 2018), CrDoCo (Chen et al., 2019) and (Toldo et al., 2020) have trained their generative model using a combination of pixel-level and feature-level adversarial losses, cycle-consistency loss and semantic-consistency loss to translate from synthetic to real domain. PixelDA (Bousmalis et al., 2017) used both noise vectors and synthetic images to train their generative model which they optimized using both adversarial and content-similarity losses. GraspGAN (Bousmalis et al., 2017) was trained on simulated data refined by a GAN network for training a robotic arm to grasp objects. For 6DoF pose estimation, (Rambach et al., 2018) translated both synthetic and real images to the common pencil filter image domain before training their network on synthetic images and evaluating it on real images. Besides these works, there have been few works addressing depth domain adaptation from synthetic to real data. We consider them in the next section.

2.3 Depth Domain Adaptation

SimGAN (Shrivastava et al., 2017) 'refined' synthetic depth images which can be then used to train models for gaze estimation or hand pose estimation on real data. (Mueller et al., 2018) adapted CycleGAN with an additional geometric consistency loss to translate depth images of hand poses from synthetic to real domain and used the resulting images for training a real-time 3D hand tracking network. Alike SimGAN, (He et al., 2019) also learned a style-transfer network to transform smooth hand pose depth images to look more realistic using a GAN. However, the source and target images used in all these methods were not as challenging containing only hand poses on clean backgrounds or eye gazes, therefore the domain difference was less significant. On the contrary, we translate from synthetic depth images of complete scenes to real depth domain. It is crucial to mention here that apart from these methods to simulate depth image noise, several works have addressed the conventional depth image enhancement tasks using image translation based domain adaptation as well (Gu et al., 2020; Agresti et al., 2019).

3 METHODS AND MATERIALS

3.1 Realistically Refining Synthetic Depth Images

We base our network that translates from synthetic depth images to real depth images on CycleGAN introduced by (Zhu et al., 2017) which uses, in addition to the standard adversarial loss, a cycle-consistency loss to ensure that the result of successive forward and reverse mappings on an input image is consistent with the original input image. We refer to this as Refiner network. The Refiner network therefore comprises two generators $G_{syn \to real}$ and $G_{real \to syn}$, paired with their corresponding domain discriminators D_{real} and D_{syn} respectively. For the synthetic$\to$real translation, $G_{syn \to real}$ represents the convolutional neural network which transforms the image such that the discriminator D_{real} maximizes the probability that the transformed image is taken from the real domain. D_{real} on the other hand is trained to assign correct probability to samples taken from the real domain and the ones generated from $G_{syn \to real}$. This is implemented as using a cross-entropy loss to train D_{real}. In parallel, $G_{real \to syn}$ and D_{syn} are trained in similar fashion. Additionally, the cycle consistency enforces that when both the transformed images in either direction are transformed back to their original

domain using the two generators, then the reverse mapping is as close to the original image as possible. We use L1 loss to measure the deviation of the translated image from the source image after consecutive source→target and target→source mapping have been applied to the source image.

CycleGAN does not require paired input images from source and target domain for training and therefore is ideal for our requirement as in our experiments we do not have access to paired synthetic and real depth images for the same scene, for example, in the form of real depth images captured using a depth sensor and synthetic depth rendering of the same scene acquired from a depth simulator. Moreover, the cycle consistency imposes a stronger constraint than a standard GAN with only reconstruction loss to bring the mapped image closer to the real image domain.

Figure 2(B) illustrates how CycleGAN fits in our image translation based domain adaptation framework. We use it for mapping images from synthetic domain to real domain and vice-versa using cycle-consistency and adversarial losses. We use the same network architecture for the generator and the discriminator as in the CycleGAN paper. All images are resized to size 512x512 and normalized to the range [-1,1] before passing them into the refiner network. We use nine residual blocks in the generator for processing our 512x512 size image due to memory limitations. We train this network for 20 epochs with a batch size of 2. We use Adam (Kingma and Ba, 2015) for optimization with an initial learning rate of 0.0002. We do not use the identity loss mentioned in the original CycleGAN paper.

3.2 Training Person Detection and Segmentation Networks

After synthetic depth images have been realistically modified by the image translation network, we pair them with the corresponding bounding box and segmentation mask annotations acquired from the synthetic dataset to train Faster R-CNN (Ren et al., 2015) and Mask R-CNN (He et al., 2017) networks for person detection and segmentation respectively. This method can be trivially extended to other task networks in the presence of given annotations. Doing so is possible because the refiner network does not semantically alter the source images and only changes the style. Our Faster R-CNN network uses VGG16 network (Simonyan and Zisserman, 2015) as its backbone and is pretrained on ImageNet dataset (Deng et al., 2009). For Mask R-CNN we adopt the ResNet-101 (He et al., 2016) network as its backbone and initialize it with weights pretrained on COCO dataset.

All images are resized to 600x600 before training to maintain aspect ratio same as the real images and shorter side same as in the original Faster R-CNN implementation. Faster R-CNN uses an initial learning rate of 0.001 while Mask R-CNN uses a learning rate of 0.005. The learning rate for both networks is decreased by a factor of 10 every 3 epochs. Both networks are trained with Stochastic Gradient Descent (SGD) optimizer and momentum of 0.9.

3.3 Dataset

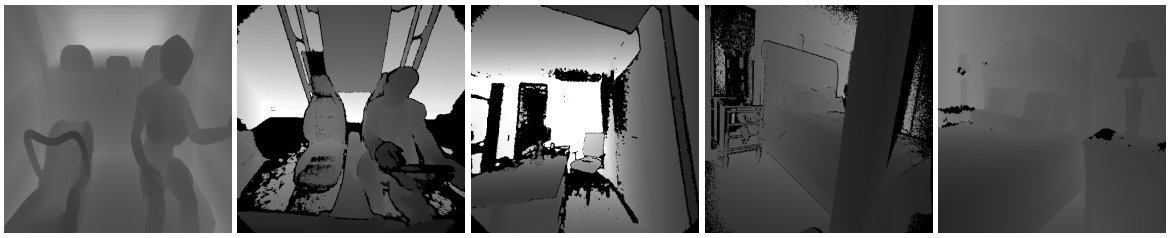
3.3.1 Synthetic dataset

In this paper, we show the efficacy of using an adversarial training based generator to realistically modify synthetic depth images and using those depth images to train person detection and segmentation networks. We demonstrate the usability of task networks trained in this manner for the use-case of driver monitoring inside a car cabin. To achieve this we used two datasets containing synthetic and real depth images of car in-cabin scenes as the source and target domain dataset respectively where the scene is viewed from the front.

We use SVIRO (Dias Da Cruz et al., 2020) which is a dataset of synthetically generated car rear in-cabin scenes and contains RGB, depth and infrared images for 10 different car models along with ground truth labels for classification, object detection, semantic segmentation and keypoint estimation tasks. We use 4,000 images from this dataset as the source domain images to train the image refiner network. In addition to it, we use 10,000 synthetic images from this dataset to train our baseline person detection and person segmentation networks which we use to compare against the same models trained on the images realistically refined by our image refiner network. This gives us the lower performance limit for our method. Figure 3a shows an example image from this dataset.

3.3.2 Real dataset

We created our own dataset of real depth images showing front-view of a car cabin using a Kinect AZURE camera fitted in the driving simulator introduced in (Feld et al., 2020) and as shown in Figure 4. We fix the camera in front of the driving simulator where the rear view mirror would be in a real car and point it slightly downwards so that the front-seat compartment is well captured. We capture sequences of 6 subjects in total where either one or two of the subjects are present in the scene. We always have at least one person in the scene driving, and in the sequences where a passenger is present in the scene, he



(a) SVIRO (b) CABIN (c) AZURE (d) Kinect V2 (e) Kinect V1

Figure 3: Example synthetic and real images used for image translation and domain adaptation.

Table 1: Overview of the size and usage of the synthetic and real datasets used in this work.

Domain	Description	Usage	images
Synthetic	Images from SVIRO	Train refiner network	4000
	Images from SVIRO	Train task networks (lower limit)	10000
Real	Captured in driving simulator with Kinect AZURE (CABIN)	Train Refiner(syn→CABIN)	4176
	Captured indoor scenes with Kinect AZURE (AZURE)	Train Refiner(syn→AZURE)	1000
	Kinect V1 images from SUN RGB-D	Train Refiner(syn→Kinect V1)	1000
	Kinect V2 images from SUN RGB-D	Train Refiner(syn→Kinect V2)	1000
	Captured in driving simulator with Kinect AZURE (CABIN)	Train task networks (upper limit)	3300
	Captured in driving simulator with Kinect AZURE (CABIN)	Evaluate all models	876

Figure 4: Data capture setup with a Kinect AZURE fixed at the front of a driving simulator.

or she is doing predefined actions like accessing the glove compartment, talking to the driver, etc. This gives us an image set of 4176 depth images to which we refer as CABIN dataset (Figure 3b). Note that unlike SVIRO we capture front seat images only. We split the 4176 images such that frames belonging to a sequence are part of either training or testing set. Splitting in this manner gives image sets of 3300 and 876 images which we use to train and test respectively the person detection and person segmentation models. The task models trained on this training set thus give the upper performance limit against which we can compare the same task models but trained on realistically refined synthetic images. The test set consisting of 876 images is used for evaluating the models trained on synthetic images, real images and realistically modified synthetic images in different experiments.

For training the image refiner network we make use of this complete dataset of 4176 images as the target domain image set. Furthermore, for one of our experiments that evaluates the benefit of using a more general image set as the target domain, we extend this real image set by 1000 images of indoor scenes captured with the same Kinect AZURE camera. We refer to this dataset containing in total 5176 images from both the driving simulator and indoor scenes captured with Kinect AZURE as AZURE dataset (Figure 3c).

Since one of our goals in this work is to study the effect of using different image sets captured using different types of depth sensors on domain adaptation, we create two separate datasets of 1000 images each containing images acquired with Kinect V1 and Kinect V2 cameras. These images are taken from the SUN RGB-D dataset (Song et al., 2015). We combine these 1000 images with the CABIN dataset to create two datasets we call Kinect V1 and Kinect V2 datasets. Figure 3d and 3e show sample images from Kinect V2 and Kinect V1 respectively. For an overview of different image sets used in this work, please refer to Table 1.

Table 2: Comparison of mAP score, and IoU and Precision scores for Faster R-CNN and Mask R-CNN networks respectively, trained on synthetic images, real images and different image sets refined by refiner network.

	Training set	mAP	IoU	Precision
Baseline	Synthetic	72.1	23.6	56.4
	Real	82.9	80.7	87.7
	Synthetic+Real(FineTuned)	**89.7**	81.7	**91.6**
Refined	Refined(syn→CABIN)	78.0	35.3	47.8
	Refined(syn→AZURE)	83.9	25.1	34.4
	Refined(syn→Kinect V2)	79.8	42.6	44.9
	Refined(syn→Kinect V1)	80.1	39.9	43.7
Finetuned	Refined(syn→AZURE)+Real(FineTuned)	89.3	**86.2**	89.8

4 EXPERIMENTS

4.1 Baseline experiments

We first conduct some experiments to establish the baselines as described in sections 3.3.1 and 3.3.2 to effectively evaluate the advantage of our domain adaptation approach over using only synthetic or only real data for training. We train three person detection networks and three corresponding person segmentation networks on the following sets of training images: a) only 10000 synthetic images from SVIRO; b) only 4176 real images from CABIN dataset and c) 10000 synthetic images from SVIRO with further fine-tuning on 4176 real images from CABIN dataset.

The first three rows of Table 2 show the mean Average Precision (mAP) of the three Faster R-CNN models trained this way, and the Intersection-over-Union (IoU) and Precision of the three Mask R-CNN models. As mentioned in section 3.3, all models are evaluated on the held-out test set of 876 real depth images. We can note that as expected the Faster R-CNN and Mask R-CNN models trained on real images perform better than the model trained on only synthetic images, and further fine-tuning these models using real images improves the performance metrics even beyond the baseline of the model trained on real images only.

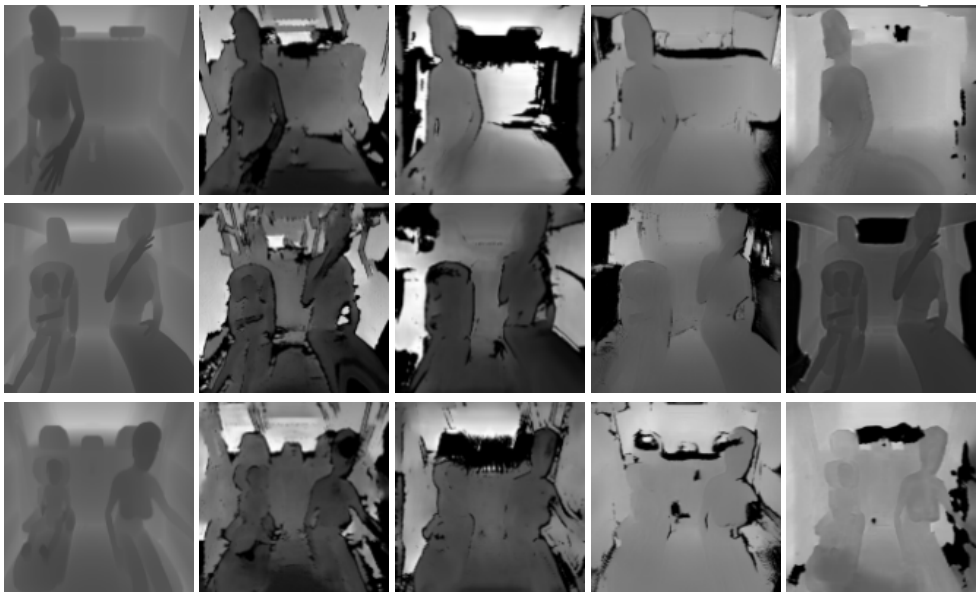
4.2 Refinement using only CABIN dataset

Keeping the model architecture and training hyperparameters same, we first perform image translation from synthetic images to real domain using only the depth images captured in the driving simulator as the target domain images. Since our ultimate goal is to perform person detection and segmentation on these images, it should be sufficient to use these images for image translation. However there is very limited variation in the background of these images, noticeably the vertical lines/bars that are present due to the physical data capturing setup. In such a case, as shown in Figure 5b, these structural elements are mistaken as the 'style' of the image and therefore while translating the image from synthetic domain to real domain, the generator introduces these structures in the translated image. We can see that despite introducing new semantics in the image in the form of background structure, the image translation network adapts the style of depth images quite well in the form of missing pixels along the edges and overall holes in the image. Moreover, the grey values of the refined image are also shifted to match the CABIN image set which means the depth values have been adapted. Although this result may be undesirable in some cases, for our target domain these images look more realistic.

Quantitatively, in the fourth row of Table 2 we can observe that training with these realistically refined images improves the performance of Faster R-CNN for person detection by a significant margin compared to when the network is trained on only synthetic image (first row). Whereas for Mask R-CNN, this improvement is seen only in the IoU metric while the precision decreases.

4.3 Refinement using AZURE dataset

Since using only real images from the driving simulator as the target domain introduces artifacts in the realistically refined images, we extend the target domain dataset by additional 1000 images capturing indoor scenes but with the same camera (Figure 3c). These images add some variability to the dataset scenery and consequent noise patterns so that the refiner network does not attribute the style of the depth images to the presence of physical structures in the scene. As seen in Figure 5c, this decreases the background artifacts in the refined images while keeping the depth-specific noise patterns. We can also observe that training with additional indoor scene data improves the mAP for object detection compared to



(a) Synthetic (b) Syn→CABIN (c) Syn→AZURE (d) Syn→Kinect V2 (e) Syn→Kinect V1

Figure 5: Qualitative comparison of out of refiner network trained using different target domain image sets.

the model trained on only CABIN data surpassing the set upper performance limit on real images. This shows that using plentiful realistically refined images for person detection in depth images is a better alternative to using limited real-world data. We think that this improved result may be explained by a bigger target domain image set of real images which has sufficient scene variation. However, surprisingly this approach hampers the performance of person segmentation network. We think it is a result of loss of details in the translated images that happens possibly due to the target domain image set comprising mixed images from car cabin and indoor scenes. This loss of image details is evident from comparing Figure 5b with 5c. Since for image segmentation such details are important, the refined image set does not serve Mask R-CNN well. This is a point for more investigation.

We also explore the fine-tuning with real data approach in this direction, meaning that instead of training the two task networks on the synthetic images we train them on their realistically refined counterparts and then fine-tune those networks on real images. As shown in the last row of Table 2, this gives a significant improvement for all performance metrics of both tasks, vastly surpassing the upper performance limit of task networks trained on real images. On top of this, the task models trained this way almost close the gap to the model trained with the Synthetic+Real(FT) strategy, while even surpassing it in the IoU metric proving that refined images are a more beneficial replacement for synthetic images.

4.4 Refinement using images from Kinect V2

Kinect V2 works on the time-of-flight principle to capture depth much like the Kinect AZURE camera, but unlike AZURE it has higher power requirements and is not as portable. Nonetheless, we would like to compare how different depth sensors working on same depth capturing principle affect the image translation output and thereby the domain adaptation results. For this we use the SUN RGB-D dataset which has depth images from various sensors working on both time-of-flight and pattern projection principles. We keep the 4196 images from the CABIN dataset and add 1000 images from SUN RGB-D which were captured using Kinect V2 sensor (Figure 3d). We perform image translation and domain adaptation similarly as in previous experiments and evaluate the results. Figure 5d shows visually the results of image translation using CycleGAN. Note that while fewer pixels are missing along the object edges, there is a heavy loss of details in the images with smaller objects and children completely blurred. Our expectation was that due to similar depth capture mechanism but difference in the sensor itself the performance of the task networks trained on these images should be lower than the networks trained on images refined using image-set created using same camera as test set. But as seen in sixth row of Table 2, this decrease is seen in only mAP and again the IoU and Precision metrics do not follow expectations.

4.5 Refinement using images from Kinect V1

Lastly, we want to compare against images refined using a different sensor working on a different depth capture principle altogether, that is, pattern projection. Similar to last section, we prepare a dataset of 4186 + 1000 images using CABIN dataset and Kinect V1 images from SUN RGB-D dataset. We then train the refiner network on this combined image set and thereafter train the task networks on the consequently refined images. As seen in Figure 3e, images captured with this camera having missing depth values or holes but do not have the missing edge pixels characteristic to time-of-flight depth sensors. Here one would expect the domain adaptation to be slightly worse than domain adaptation with Kinect V2 images since the depth sensors works on a different principle. As evident from last column of Figure 5, images refined this way do not exhibit any missing values close to object border in the images. Instead they show only large holes in the images similar to what is found in depth images captured using a pattern projection depth sensor (Figure 3e). We can also note that these refined images lose semantic details to a much lesser extent compared to images refined using any of the other time-of-flight sensor image sets. This is perhaps why the performance of Mask R-CNN on image segmentation metrics is quite high compared to Refined(syn→AZURE) image set. The mAP metric shows negligible improvement compared to Refined(syn→Kinect V2) image set but is in the end inferior to the image set refined using same depth sensor as the target image set on which evaluation is performed, that is, Refined(syn→AZURE).

5 CONCLUSION AND FUTURE WORK

In this paper, we studied one of the less frequently explored methods for domain adaptation, namely Adversarial Image Translation based Domain Adaptation. We showed that in absence of paired source and target domain images, one can resort to unsupervised image translation to first realistically modify synthetic depth images to look as if they come from a real depth sensor and then use them for training final task networks. More specifically we demonstrate the viability of this approach for person detection and segmentation tasks in real depth images captured with a Kinect AZURE inside a car-cabin. We saw how we can stretch the potential of this approach using standard fine-tuning strategies where synthetic data is replaced with data refined by our image refiner network. Through visual analysis of the refined images we confirmed that it is possible to generate visually convincing 'realistically refined' images that mimic the noise patterns of the real depth images acquired by a depth sensor. Lastly, we demonstrated experimentally that the choice of the depth sensor used to capture target domain image set for image translation affects the end result of domain adaptation. Using same depth sensor or similar kind of depth sensor improves performance of domain adaptation for person detection task. Whereas for person segmentation, it is more important to use a target domain image set which does not cause loss of details during image refinement. We leave it as future work to improve the image refiner network to minimize loss of image details and to validate this approach on other larger depth datasets.

ACKNOWLEDGEMENTS

This work was partially funded within the Electronic Components and Systems for European Leadership (ECSEL) Joint Undertaking in collaboration with the European Union's H2020 Framework Program and Federal Ministry of Education and Research of the Federal Republic of Germany (BMBF), under grant agreement 16ESE0424 / GA826600 (VIZTA).

REFERENCES

Agresti, G., Schaefer, H., Sartor, P., and Zanuttigh, P. (2019). Unsupervised domain adaptation for tof data denoising with adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., and Krishnan, D. (2017). Unsupervised pixel-level domain adaptation with generative adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104.

Chen, Y., Lin, Y., Yang, M., and Huang, J. (2019). Crdoco: Pixel-level domain transfer with cross-domain consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Dias Da Cruz, S., Wasenmüller, O., Beise, H., Stifter, T., and Stricker, D. (2020). Sviro: Synthetic vehicle interior rear seat occupancy dataset and benchmark. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Dundar, A., Liu, M., Wang, T., Zedlewski, J., and Kautz, J. (2018). Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation. *ArXiv*, abs/1807.09384.

Feld, H., Mirbach, B., Katrolia, J., Selim, M., Wasenmüller, O., and Stricker, D. (2020). Dfki cabin simulator: A test platform for visual in-cabin monitoring functions. In *Commercial Vehicle Technology 2020 - Proceedings of the 6th Commercial Vehicle Technology Symposium - CVT 2020*.

Ghifary, M., Kleijn, B., and Zhang, M. (2014). Domain adaptive neural networks for object recognition. In Pham, D.-N. and Park, S.-B., editors, *PRICAI 2014: Trends in Artificial Intelligence*, pages 898–904, Cham. Springer International Publishing.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Gu, X., Guo, Y., Deligianni, F., and Yang, G. (2020). Coupled real-synthetic domain adaptation for real-world deep depth enhancement. *IEEE Transactions on Image Processing*, 29:6343–6356.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

He, W., Zhongzhao, X., Li, Y., Wang, X., and Cai, W. (2019). Synthesizing depth hand images with gans and style transfer for hand pose estimation. *Sensors*, 19:2919.

Hoffman, J., Tzeng, E., Darrell, T., and Saenko, K. (2017). *Simultaneous Deep Transfer Across Domains and Tasks*, pages 173–187.

Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). Cycada: Cycle-consistent adversarial domain adaptation. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR 80:1989-1998.

Hu, J., Lu, J., and Tan, Y. (2015). Deep transfer metric learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 325–333.

Isola, P., Zhu, J., Zhou, T., and Efros, A. (2017). Image-to-image translation with conditional adversarial networks. *CVPR*.

Karacan, L., Akata, Z., Erdem, A., and Erdem, E. (2016). Learning to generate images of outdoor scenes from attributes and semantic layouts.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Li, Y., Liu, M., Li, X., Yang, M., and Kautz, J. (2018). A closed-form solution to photorealistic image stylization. In *ECCV*.

Liu, M.-Y., Breuel, T., and Kautz, J. (2017). Unsupervised image-to-image translation networks. *ArXiv*, abs/1703.00848.

Motiian, S., Piccirilli, M., Adjeroh, D. A., and Doretto, G. (2017). Unified deep supervised domain adaptation and generalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5716–5726.

Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., and Theobalt, C. (2018). Ganerated hands for real-time 3d hand tracking from monocular rgb. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 49–59.

Rambach, J., Deng, C., Pagani, A., and Stricker, D. (2018). Learning 6dof object poses from synthetic single channel images. In *Proceedings of the 17th IEEE ISMAR —. IEEE International Symposium on Mixed and Augmented Reality (ISMAR-2018), 17th, October 16-20, München, Germany*. IEEE, IEEE.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 91–99, Cambridge, MA, USA. MIT Press.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2242–2251.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Song, S., Lichtenberg, S. P., and Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576.

Toldo, M., Michieli, U., Agresti, G., and Zanuttigh, P. (2020). Unsupervised domain adaptation for mobile semantic segmentation based on cycle consistency and feature alignment. *Image Vision Comput.*, 95(C).

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *ArXiv*, abs/1412.3474.

Zhu, J., Park, T., Isola, P., and Efros, A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. pages 2242–2251.