AugGAN: Cross Domain Adaptation with GAN-Based Data Augmentation

Sheng-Wei Huang[1(✉)], Che-Tsung Lin[1,2], Shu-Ping Chen[1], Yen-Yi Wu[1], Po-Hao Hsu[1], and Shang-Hong Lai[1]

[1] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
shengwei@mx.nthu.edu.tw, lai@cs.nthu.edu.tw

[2] Intelligent Mobility Division, Mechanical and Mechatronics Systems Research Laboratories, Industrial Technology Research Institute, Zhudong, Taiwan
AlexLin@itri.org.tw

Abstract. Deep learning based image-to-image translation methods aim at learning the joint distribution of the two domains and finding transformations between them. Despite recent GAN (Generative Adversarial Network) based methods have shown compelling results, they are prone to fail at preserving image-objects and maintaining translation consistency, which reduces their practicality on tasks such as generating large-scale training data for different domains. To address this problem, we purpose a structure-aware image-to-image translation network, which is composed of encoders, generators, discriminators and parsing nets for the two domains, respectively, in a unified framework. The purposed network generates more visually plausible images compared to competing methods on different image-translation tasks. In addition, we quantitatively evaluate different methods by training Faster-RCNN and YOLO with datasets generated from the image-translation results and demonstrate significant improvement on the detection accuracies by using the proposed image-object preserving network.

Keywords: Generative adversarial network
Image-to-image translation · Semantic segmentation
Object detection · Domain adaptation

1 Introduction

Deep learning pipelines have stimulated substantial progress for general object detection. Detectors kept pushing the boundaries on several detection datasets.

S.-W. Huang and C.-T. Lin—Indicates equal contribution.

The original version of this chapter was revised: The presentation of Figure 1 was updated. The correction to this chapter is available at https://doi.org/10.1007/978-3-030-01240-3_50

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01240-3_44) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11213, pp. 731–744, 2018.
https://doi.org/10.1007/978-3-030-01240-3_44

However, despite being able to efficiently detect objects seen by arbitrary viewing angles, CNN-based detectors are still limited in a way that they could not function properly when faced with domains significantly different from those in the original training dataset. The most common way to obtain performance gain is to go through the troublesome data collection/annotation process. Nevertheless, the recent successes of Generative Adversarial Networks (GANs) on image-to-image translation have opened up possibilities in generating large-scale detection training data without the need for object annotation.

Generative adversarial networks [1], which put two networks (i.e., a generator and a discriminator) competing against each other, have emerged as a powerful framework for learning generative models of random data distributions. While expecting GANs to produce an RGB image and its associated bounding boxes from a random noise vector still sounds like a fantasy, training GANs to translate images from one scenario to another could help skip the tedious data annotation process. In the past, GAN-based image-to-image translation methods, such as Pix2Pix [2], were considered to have limited applications due to the requirement for pairwise training data. Although these methods yielded impressive results, the fact that they require pairwise training images largely reduces their practicality for the problem that we aim to solve.

Recently, unpaired image-to-image translation methods have achieved astonishing results on various domain adaptation challenges. Having almost identical architectures, CycleGAN [3], DiscoGAN [4], and DualGAN [5] made unpaired image-to-image translation possible through introducing the cycle consistency constraint. CoGAN [6] is a model which also works on unpaired images, using two shared-weight generators to generate images of two domains with one random noise. UNIT [7] is an extension of CoGAN. Aside from having similar hard weight-sharing constraints as CoGAN, Liu et al. further implemented the latent space assumption by encouraging two encoders to map images from two domains into the same latent space, which largely increases the translation consistency. These methods all demonstrate compelling visual results on several image-to-image translation tasks; however, what hinders the capability of these methods for providing large-scale detection training data, specifically when faced with translation tasks with a large domain shift, is the fact that these networks often arrive at solutions where the translation results are indistinguishable from the the target domain in terms of style, and usually contain corrupted image-objects.

In this paper we propose a structure-aware image-to-image translation network, which allows us to directly benefit object detection by translating existing detection RGB data from its original domain other scenarios. The contribution of this work is three-fold: (1) We train the encoder networks to extract structure-aware information through the supervision of a segmentation subtask, (2) we experiment on different weight sharing strategy to ensure the preservation of image-objects during image-translations, and (3) our object-preserving network provides significant performance gain on the night-time vehicle detection.

We stress particularly on day-to-night image translation not only for the importance of night-time detection, but also for the fact that day/night image

translation is one of the most difficult domain transformations. However, our method is also capable of handling various domain pairs. We train our network on synthetic (i.e., SYNTHIA [8], GTA dataset [9]) Compared to the competing methods, the domain translation results of our network significantly enhance the capability of the object detector for application on both synthetic (i.e., SYNTHIA, GTA) and real-world (i.e., KITTI [10], ITRI) data. In addition, we welcome those who are interested in the ITRI dataset to email us for provision.

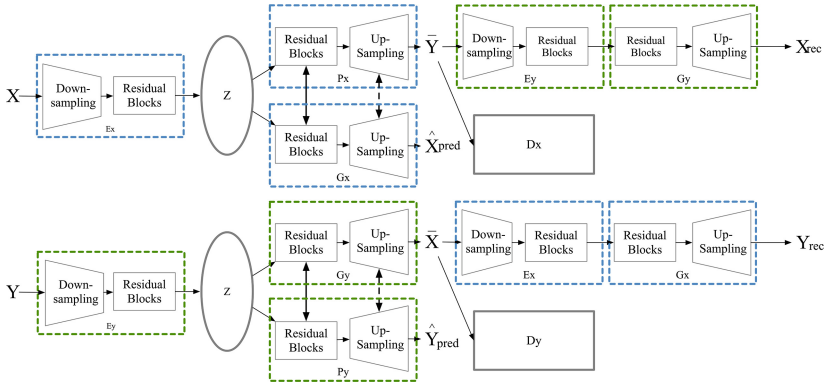

Fig. 1. Overall structure of the proposed image-to-image translation network. X, Y: image domain X and Y; Z: feature domain; $\hat{X}_{pred}$, $\hat{Y}_{pred}$: predicted segmentation masks; $\bar{X}$, $\bar{Y}$: translated results; dotted line implicates soft-sharing, solid line implicates hard-sharing.

2 Proposed Framework

In unsupervised image-to-image translation, models learn joint distribution where the network encodes images from the two domains into a shared feature space. We assume that, for an image to be properly translated to the other domain, the encoded information is required to contain (1) mutual style information between domain A and B, and (2) structural information of the given input image, as illustrated in Fig. 1. Based on the assumption we design our network to jointly optimize image-translation and semantic segmentation. Through our weight-sharing strategy, the segmentation subtask serves as an auxiliary regularization for image-translation.

Let X and Y denote the two image domains, $\hat{X}$ and $\hat{Y}$ denote the corresponding segmentation masks, and Z represent the encoded feature space. Our network, as depicted in Fig. 1, consists of two encoders $E_x : \mathrm{X} \rightarrow \mathrm{Z}$ and $E_y : \mathrm{Y} \rightarrow \mathrm{Z}$, two generators, $G_x : \mathrm{Z} \rightarrow \bar{\mathrm{Y}}$ and $G_y : \mathrm{Z} \rightarrow \bar{\mathrm{X}}$, two segmentation generators, $P_x : \mathrm{Z} \rightarrow \hat{\mathrm{X}}_{\mathrm{pred}}$, and $P_y : \mathrm{Z} \rightarrow \hat{\mathrm{Y}}_{\mathrm{pred}}$, and two discriminators D_x and D_y for the two image domains, respectively. Our network learns image domain translation in both directions and the segmentation sub-tasks simultaneously. For an input

$x \in X$, E_x first encodes x into the latent space, and the 256-channel feature vector is then processed to produce (1) the translated output $\bar{y}$ via G_x, and (2) the semantic representation $\hat{x}_{pred}$ via P_x. The translated output $\bar{y}$ is then fed through the inverse encoder-generator pair $\{E_y, G_y\}$ to yield the reconstructed image x_{rec}. Detailed architecture of our network is given in Table 1.

Table 1. Network architecture for the image-to-image translation experiments. N, K, and S denote the number of convolution filters, kernel size, and stride, respectively

Layer	Encoders	Layer info
1	CONV	N64, K7, S1
2	CONV, ReLU	N128, K3, S2
3	CONV, ReLU	N256, K3, S2
4	RESBLK, ELU	N512, K3, S1
5	RESBLK, ELU	N512, K3, S1
5	RESBLK, ELU	N512, K3, S1
Layer	Generators/parsing networks	Layer info
1	RESBLK, ELU	N512, K3, S1, hard shared
2	RESBLK, ELU	N512, K3, S1, hard shared
3	RESBLK, ELU	N512, K3, S1, hard shared
4	RESBLK, ELU	N512, K3, S1, hard shared
5	RESBLK, ELU	N512, K3, S1, hard shared
6	RESBLK, ELU	N512, K3, S1, hard shared
7	DCONV, ReLU	N128, K3, S2, soft shared
8	DCONV, ReLU	N64, K3, S2, soft shared
9 (generator)	CONV, Tanh	N3, K7, S1
9 (parsing net)	CONV, ReLU	N (task specific), K7, S1
10 (parsing net)	CONV, Softmax	N6 (task specific), K1, S0
Layer	Discriminator	Layer info
1	CONV, LeakyReLU	N64, K4, S2
2	CONV, LeakyReLU	N128, K4, S2
3	CONV, LeakyReLU	N256, K4, S2
3	CONV, LeakyReLU	N512, K4, S2
3	CONV, LeakyReLU	N512, K4, S1
3	CONV, Sigmoid	N1, K4, S1

2.1 Structure-Aware Encoding and Segmentation Subtask

We actively guide the encoder networks to extract context-aware features by regularizing them via segmentation subtask so that the extracted 256-channel feature vector contains not only mutual style information between X and Y

domains, but also the intricate low-level semantic features of the input image that are valuable in the preservation of image-objects during translation. The segmentation loss is formulated as:

$$\begin{aligned}\mathcal{L}_{seg-x}(P_x, E_x, X, \hat{X}) = \lambda_{seg-L1}\mathbb{E}_{x\sim p_{data(x)}}[\|P_x(E_x(x)) - \hat{x}\|_1] \\ + \lambda_{seg-crossentropy}\mathbb{E}_{x\sim p_{data(x)}}[\|\log(P_x(E_x(x)) - \hat{x})\|_1]\end{aligned} \tag{1}$$

$$\begin{aligned}\mathcal{L}_{seg-y}(P_y, E_y, Y, \hat{Y}) = \lambda_{seg-L1}\mathbb{E}_{y\sim p_{data(y)}}[\|P_y(E_y(y)) - \hat{y}\|_1] \\ + \lambda_{seg-crossentropy}\mathbb{E}_{y\sim p_{data(y)}}[\|\log(P_y(E_y(y)) - \hat{y})\|_1]\end{aligned} \tag{2}$$

2.2 Weight Sharing for Multi-task Network

Sharing weights between the generator and parsing network allows the generator to fully take advantage of the context-aware feature vector. We hard-share the first 6 residual blocks and soft-share the subsequent two deconvolution blocks for generators and parsing networks. We experiment on different weight-sharing strategies, as illustrated in Sect. 3.2, such as hard-share, not sharing the deconvolution blocks, and not sharing the residual blocks, and come to the best sharing strategy. We calculate the weight difference between deconvolution layers of the two networks and model the difference as a loss function through mean square error with target as a zero matrix. The mathematical expression for the soft weight sharing loss function is given by

$$\mathcal{L}_{\omega}(\omega_G, \omega_P) = -\log((\omega_{G_x} \cdot \omega_{P_x}/\|\omega_{G_x}\|_2\|\omega_{P_x}\|_2)^2) \tag{3}$$

where ω_G and ω_P denote the weight vectors formed by the deconvolution layers of the generator and parsing networks, respectively.

2.3 Cycle Consistency

The cycle consistency loss has been proven quite effective in preventing network from generating random images in the target domain. We also enforce the cycle-consistency constraint in the proposed framework to further regularize the ill-posed unsupervised image-to-image translation problem. The loss function is given by

$$\begin{aligned}\mathcal{L}_{cyc}(E_x, G_x, E_y, G_y, X, Y) = \mathbb{E}_{x\sim p_{data(x)}}[\|G_y(E_y(G_x(E_x(x)))) - \mathrm{x}\|_1] \\ + \mathbb{E}_{y\sim p_{data(y)}}[\|G_x(E_x(G_y(E_y(y)))) - y\|_1].\end{aligned} \tag{4}$$

2.4 Adversarial Learning

Our network contains two Generative Adversarial Networks: GAN_1: $\{E_x, G_x, D_x\}$, and GAN_2: $\{E_y, G_y, D_y\}$. We apply adversarial losses to both GANs, and formulate the objective loss functions as:

$$\begin{aligned}\mathcal{L}_{GAN_1}(E_x, G_x, D_x, X, Y) = \mathbb{E}_{y\sim p_{data(y)}}[\log D_x(y)] \\ + \mathbb{E}_{x\sim p_{data(x)}}[\log(1 - D_x(G_x(E_x(x))))]\end{aligned} \tag{5}$$

$$\begin{aligned}\mathcal{L}_{GAN_2}(E_y, G_y, D_y, Y, X) = \mathbb{E}_{x\sim p_{data(x)}}[\log D_y(x)] \\ + \mathbb{E}_{y\sim p_{data(y)}}[\log(1 - D_y(G_y(E_y(y))))]\end{aligned} \tag{6}$$

2.5 Network Learning

We jointly solve the learning problems for the image-translation streams: $\{E_1, G_1\}$ and $\{E_2, G_2\}$, the image-parsing streams: $\{E_1, P_1\}$ and $\{E_2, P_2\}$, and two GAN networks: GAN_1 and GAN_2, for training the proposed network. The integrated objective function is given as follows:

$$\begin{aligned}\mathcal{L}_{full} = &\mathcal{L}_{GAN}(E_x, G_x, D_x, X, Y) + \mathcal{L}_{GAN}(E_y, G_y, D_y, Y, X) \\ &+ \lambda_{cyc} * \mathcal{L}_{cyc}(E_x, G_x, E_y, G_y, X, Y) \\ &+ \lambda_{seg} * (\mathcal{L}_{seg}(E_x, P_x, X, \hat{X}) + \mathcal{L}_{seg}(E_y, P_y, Y, \hat{Y})) \\ &+ \lambda_{\omega} * (\mathcal{L}_{\omega_x}(\omega_{G_x}, \omega_{P_x}) + \mathcal{L}_{\omega_y}(\omega_{G_y}, \omega_{P_y}))\end{aligned} \tag{7}$$

3 Experimental Results

Though many works were dedicated on providing large-scale vehicle datasets for the research community [11–15], most public are collected in daytime. Considering that CNN-based detectors highly rely data augmentation techniques to stimulate performance, training detectors with both day and night images is necessary so as to make them more general. Synthetic dataset, such as SYNTHIA or GTA dataset, provides diverse on-road synthetic sequences as well as segmentation masks in scenarios such as day, night, snow, etc. As our network requires both segmentation mask and nighttime image, we conducted the training of our network with SYNTHIA and GTA datasets. For evaluation purpose, however, we utilize real-world data such as KITTI and our ITRI datasets.

The performance of the network was further analyzed through training YOLO [16] and Faster R-CNN (VGG 16-based) [17] detectors with generated image sets. Aside from revising both detectors to perform 1-class vehicle detection, all hyper-parameters were the same as those used for training on PASCAL VOC challenge. The IOU threshold for objects to be considered true-positives is 0.5, where we follow the standard for common object detection datasets. In the transformation of segmentation Ground-Truth to its counterpart in detection, we exclude the bounding boxes whose heights lower than 40 pixels or occluded for more than 75% in the subsequent AP estimation.

3.1 Synthetic Datasets

We first assess the effectiveness of training detectors with transformed images in both day and night scenarios. We evaluated our network, which is trained with SYNTHIA, by training detectors with transformed images produced by our network. As shown in Table 2, AugGAN outperforms competing methods in both day and night scenarios. AugGAN also surpasses its competitors when trained with GTA dataset, see Table 3. Visually, the transformation results of AugGAN is clearly better in terms of image-object preservation and preventing the appearance of artifacts as shown in Figs. 2 and 3.

Fig. 2. SYNTHIA day-to-night transformation results - GANs trained with SYNTHIA: first row: SYNTHIA daytime testing images; second row: results of CycleGAN; 3rd row: results of UNIT; 4th row: results of AugGAN

Fig. 3. GTA day-to-night transformation results - GANs trained with GTA: first row: GTA daytime testing images; second row: outputs of CycleGAN; 3rd row: outputs of UNIT; 4th row: outputs of AugGAN.

Table 2. Detection accuracy comparison (AP) - GANs trained with SYNTHIA. SDTrain/SNTrain: SYNTHIA daytime/nightime training set; SDTest/SNTest: SYNTHIA daytime/nighttime testing set.

Training	Testing	CycleGAN	UNIT	AugGAN	Detector
SDTrain	SNTest	36.1	35.2	39.0	YOLO
SNTrain	SDTest	33.8	32.6	38.0	YOLO
SDTrain	SNTest	65.9	57.2	72.2	Faster RCNN
SNTrain	SDTest	65.7	62.7	70.1	Faster RCNN

Table 3. Detection accuracy comparison (AP) - detectors trained with transformed images produced by GANs (trained with GTA dataset), and tested with real images. GTA-D-Train: transformed data with GTA training daytime images as input; GTA-N-Test: GTA testing nighttime data.

Training	Testing	CycleGAN	UNIT	AugGAN	Detector
GTA-D-Train	GTA-N-Test	20.5	23.6	25.3	YOLO
GTA-D-Train	GTA-N-Test	54.4	62.5	67.4	Faster-RCNN

3.2 KITTI and ITRI-Night Datasets

Aside from testing on SYNTHIA and GTA datasets, we also assess the capability of our network on real world data, such as KITTI, which has been widely used in assessing the performance of on-road object detectors used in autonomous driving systems. With the previously trained AugGAN, be it trained with SYNTHIA or GTA dataset, we transformed the KITTI dataset (7481 images with 6686 of which contains vehicle instances) [18] to its nighttime version and evaluate the translation results via detector training. We trained vehicle detectors with the translated KITTI dataset and tested on our ITRI-Night testing set (9366 images with 20833 vehicle instances). As experimental result indicates, real-world data transformed by AugGAN quantitatively and visually achieves better result even though AugGAN was trained with synthetic dataset, see Table 4, Figs. 4 and 5.

Table 4. Detection accuracy comparison (AP) - detectors trained with transformed images produced by GANs (trained with GTA dataset and SYNTHIA), and tested with real images. KITTI-D2N-S/KITTI-D2N-G: KITTI day-to-night training data generated by GANs; ITRIN: ITRI-Night dataset.

Training	Testing	CycleGAN	UNIT	AugGAN	Detector
KITTI-D2N-S	ITRIN	20.2	19.0	31.5	YOLO
KITTI-D2N-G	ITRIN	28.5	20.5	46.0	YOLO
KITTI-D2N-S	ITRIN	59.6	49.2	65.6	Faster RCNN
KITTI-D2N-G	ITRIN	72.0	64.0	79.3	Faster RCNN

3.3 ITRI Daytime and Nighttime Datasets

We collected a set of real-driving daytime (25104 images/87374 vehicle instances) dataset, captured mostly in the same scenario as its our nighttime dataset (9366 images with 20833 vehicle instances). In Table 5, the experiments demonstrate similar results as in other datasets. The transformed day-to-night training images

Table 5. Detection accuracy comparison (AP) - detectors trained with transformed images produced by GANs (trained with SYNTHIA/GTA dataset). ITRID-D2N-S/ITRID-D2N-G: ITRI-day day-to-night training data generated by GANs trained with SYNTHIA/GTA datasets; ITRIN: ITRI-Night dataset.

Training	Testing	CycleGAN	UNIT	AugGAN	Detector
ITRID-D2N-S	ITRIN	35.5	41.3	45.3	YOLO
ITRID-D2N-G	ITRIN	37.9	42.6	44.1	YOLO
ITRID-D2N-S	ITRIN	72.4	74.5	81.2	Faster RCNN
ITRID-D2N-G	ITRIN	86.2	85.9	86.1	Faster RCNN

Fig. 4. KITTI day-to-night transformation results - GANs trained with SYNTHIA: first row: KITTI images; second row: result of CycleGAN; 3rd row: result of UNIT; 4th row: result of AugGAN.

Fig. 5. KITTI dataset day-to-night transformation results - GANs trained with GTA dataset: first row: input images from KITTI dataset; second row: outputs of CycleGAN; 3rd row: outputs of UNIT; 4th row: outputs of AugGAN

Fig. 6. ITRI-Day dataset day-to-night transformation results - GANs trained with SYNTHIA: First row: input images from ITRI-Day dataset; Second row: outputs of cycleGAN; 3rd row: outputs of UNIT; 4th row: outputs of AugGAN

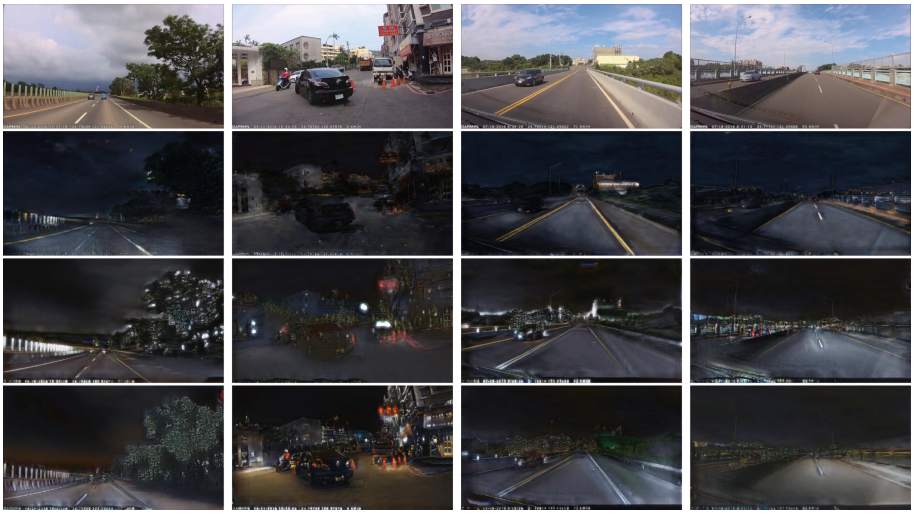

Fig. 7. ITRI-Day dataset day-to-night transformation results - GANs trained with GTA dataset: first row: input images from ITRI-Day dataset; second row: outputs of cycleGAN; 3rd row: outputs of UNIT; 4th row: outputs of AugGAN

are proved to be helpful in vehicle detector training. Training images generated by AugGAN outperforms those by competing methods due to its preservation in image-objects, with some examples shown in Figs. 6 and 7.

3.4 Transformations Other Than Daytime and Nighttime

AugGAN is capable of learning transformation across unpaired synthetic and real domains and only segmentation supervision in domain-A is required. This increases the flexibility of learning cross-domain adaptation for subsequent detector training. As shown in Fig. 8: 2nd row, our method could learn image translation from not only synthetic-synthetic, but also synthetic-real domain pairs.

Fig. 8. More image translation cases: 1st column: GTA-day to SYNTHIA; 2nd column: GTA-day to GTA-sunset; 3rd column: GTA-day to GTA-rain; 4th column: SYNTHIA-day to ITRI-night

4 Model Analysis

4.1 Segmentation Subtask

In our initial experiment on introducing the segmentation subtask, the parsing network was only utilized in the forward cycle (e.g., only day-to-night). We later on discovered that our results are improved by utilizing the parsing network to regularize both forward and inverse cycles. As can be seen in Table 6, it is quite obvious that adding regularization to the inverse cycle leads to better transformation results which make detectors more accurate. Although using only single-sided segmentation has already outperformed the previous works, introducing segmentation in both forward and backward cycles brings further accuracy improvement for object detection.

Table 6. Detection accuracy comparison (AP) - detectors trained with transformed data produced by GANs (trained with SYNTHIA). SDTrain: SYNTHIA daytime training set, transformed into nighttime; SNTest: SYNTHIA nighttime testing set.

Training	Testing	CycleGAN	UNIT	AugGAN-1	AugGAN-2	Detector
SDTrain	SNTest	36.1	35.2	38.1	39.0	YOLO
SDTrain	SNTest	65.9	57.2	68.7	72.2	Faster RCNN

4.2 Weight-Sharing Strategy

Our network design is based on the assumption that extracted semantic segmentation features of individual layers, through proper weight sharing, can serve as auxiliary regularization for image-to-image translation. Thus finding the proper weight sharing policy came to be the most important factor in our design. Weighting sharing mechanism in neural networks can be roughly categorized into soft weight-sharing and hard weight-sharing. Soft weight-sharing [19] was originally proposed for regularization and could be applied to network

Table 7. Weight-sharing strategy comparison: λ_w denotes the cosine similarity loss multiplier, with $\lambda_w = 0.02$ yielded best result. The matrix in this table is the average precision of Faster RCNN

Training	Testing	Weight-sharing strategy	AP - AugGAN
SDTrain	SNTest	Encoder: hard	39.9
SDTrain	SNTest	Encoder: hard; Decoder: hard	57.2
SDTrain	SNTest	Encoder: hard; Decoder: soft ($\lambda_w = 0.02$)	68.7

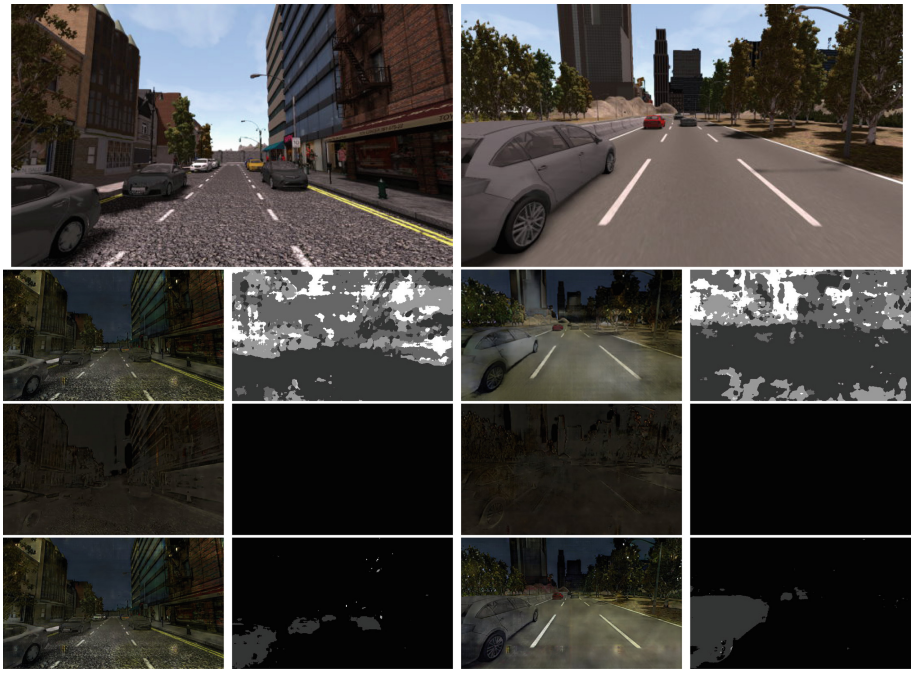

Fig. 9. Style transfer and segmentation results for different weight-sharing strategies: 1st row: input images; 2nd row: style transfer and segmentation results of hard weight sharing, hard-weighting on encoder only ($\lambda_w = 0$), and hard weighting sharing in encoder with soft-weight sharing ($\lambda_w = 0.02$) in decoder.

compression [20]. Recently, hard weight-sharing has been proven useful in generating images with similar high-level semantics [6]. The policy that we currently adopt is two-folded: (1) hard-share encoders and residual blocks of the generator-parsing net pairs, (2) soft-share deconvolution layers of the generator-parsing net pairs. We came to this setting based on extensive trial and error, and during the process we realized that both policies are integral for the optimization of our network. Without hard-sharing the said layers in (1), image-objects tends to be distorted; Without (2), the network tends to only optimize one of the tasks, see Table 7 and Fig. 9. In short, our network surpasses competing methods because our multi-task network can maintain realistic transformation style as well as preserving image-objects with the help of segmentation subtask.

5 Conclusion and Future Work

In this work, we proposed an image-to-image translation network for generating large-scale trainable data for vehicle detection algorithms. Our network is especially adept in preserving image-objects, thanks to the extra guidance of the segmentation subtask. Our method, though far from perfect, quantitatively surpasses competing methods for stimulating vehicle detection accuracy. In the future, we will continue to experiment on different tasks based on this framework, and our pursuit for creating innovative solutions for the world will continue to stride.

References

1. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS (2014)
2. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2017)
3. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)
4. Kim, T., Cha, M., Kim, H., Lee, J., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192 (2017)
5. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: unsupervised dual learning for image-to-image translation. arXiv preprint (2017)
6. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In: NIPS (2016)
7. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: NIPS (2017)
8. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR (2016)
9. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: ground truth from computer games. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 102–118. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_7
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: CVPR (2012)

11. Sivaraman, S., Trivedi, M.M.: A general active-learning framework for on-road vehicle recognition and tracking. IEEE Trans. Intell. Transp. Syst. **11**(2), 267–276 (2010)
12. Zhou, Y., Liu, L., Shao, L., Mellor, M.: DAVE: a unified framework for fast vehicle detection and annotation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 278–293. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_18
13. Yang, L., Luo, P., Change Loy, C., Tang, X.: A large-scale car dataset for fine-grained categorization and verification. In: CVPR (2015)
14. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: ICCV Workshops (2013)
15. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. IJCV **88**, 303 (2010)
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR (2016)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
18. Xiang, Y., Choi, W., Lin, Y., Savarese, S.: Subcategory-aware convolutional neural networks for object proposals and detection. In: WACV (2017)
19. Nowlan, S.J., Hinton, G.E.: Simplifying neural networks by soft weight-sharing. Neural Comput. **4**(4), 473–493 (1992)
20. Ullrich, K., Meeds, E., Welling, M.: Soft weight-sharing for neural network compression. arXiv preprint arXiv:1702.04008 (2017)