Make Virtual World Real
Final project - Advanced Topic in Deep Learning

Joanna (Ching-Hui) Hsu
Columbia University
ch3230@columbia.edu

Abstract

Modern computer vision techniques today typically require large amount of training data with accurate annotation, which is time consuming and costly to obtain. An alternative approach to have huge amount of labeled data is to use synthetic images. However, the problem is that the synthetic images may not be realistic enough, resulting in poor generalization on real test images. In this work, we investigate several methods that uses adversarial network, with synthetic images as inputs instead of random vectors. The goal is to learn a mapping from synthetic domain X to real domain Y in the absence of paired examples. We make several key modifications to the Simulated+Unsupervised (S+U) learning and Cycle-Consistent Adversarial Networks algorithm in order to explore the tasks that require geometric changes and do efficient training on images with complex foreground objects: (i) fusing local and global discriminator loss. (ii) applying dropout to the input of all convolutions in the discriminator's "global" path. (iii) using encoder-decoder network for generator. Quantitative results are presented on road segmentation task.

1. Introduction

Deep learning has rapidly transformed the state-of-the-art algorithms used to address a variety of problems in computer vision. These tasks normally require massive amounts of human annotated training data. However, labeling such large dataset is expensive and time-consuming, which has begun an impedance of deep learning progress. Thus, the idea of training on synthetic images has become appealing because the annotations are automatically available. Multi-object tracking (MOT) [9], semantic scene understanding, object detection [8] and human pose estimation [16] have been tackled using synthetic data. However, learning from synthetic images can be problematic due to gap between synthetic and real images distribution, leading the network to learn features only present in synthetic images and failing to generalize well on real images. Recently, Shrivastave *et al*. [17] has proposed Simulated+Unsupervised (S+U) learning to make the synthetic images look more realistic. They have proved that the improved realism enables the training of better models on large datasets without any human annotation effort. However, they only demonstrated the idea on images with relatively simple background, e.g NYU hand pose dataset [20] is pre-processed with background removed and the object is centered.

Figure 1. Given the complex synthetic images *(left)*, the algorithm learns to automatically improve its realism using unlabeled real data, producing the refined images *(right)*.

This paper describes a method to improve the realism of complex synthetic images from a simulation engine, such as road scene. We incorporate the idea from both Simulated+Unsupervised (S+U) learning, which the authors name it as SimGAN [17] and Cycle-Consistent Adversarial Networks, also known as CycleGAN [22] to refine synthetic images from a simulator.

This task is similar to an image-to-image translation, converting an image from one domain X to another domain Y, with unpaired training set, as opposed to pix2pix [7] which uses pairwise data.The motivation is hoping to find an unsupervised way to find image relations. Although it is not supervised, we can still exploit supervision at the level of sets: two sets in different domain are given, X and Y, so we may train a mapping from $G : X \rightarrow Y$ such that $\hat{y} = G(x), x \in X$, is indistinguishable from images $y \in Y$,

which means the distribution over $\hat{y}$ is identical to the distribution over y.

Our method in this work can be summarized as follow:

- We refine the synthetic image with a generator network that minimize the combination of adversarial loss and a self-regularization term.
- The discriminator architecture is PatchGAN [7] with additional layers added for global loss.
- To stabilize GAN training, we use Least square GAN and replay buffer.

We compare against previous approaches both qualitatively and quantitatively. The quantitative evaluation is done by conducting road segmentation using MultiNet [19]. Our code is available at https://github.com/CHJoanna/SimGan-simple.

1.1. Related Work

Generative Adversarial Networks (GANs)
The standard GAN model [4] take random Gaussian noise z, encodes into features and generates images. It learns two networks, a generator and a discriminator, with competing losses. The *adversarial loss* forces the generated images to be indistinguishable from the real images. There are many interesting and impressive application using a variety of GANs, such as image editing (iGan) [5], text2image [14], etc. We adopt the adversarial loss to learn the mapping, but make a slight modification to these models to fit our task: the model We use takes in image as input instead of noise.

Unpaired Image-to-Image Translation
Several methods have proposed to tackle the unpaired translation, where the task is to discover relations between different domains. CoGAN by Liu *et al*. [11] uses a weight-sharing strategy that favors the joint distribution domain. CycleGAN [22] and DiscoGAN [10] leverage cycle consistency loss which relies on transitivity to regularize the transfer functions. SimGAN [17] encourages the input and output to share certain content features such as image pixel space. The regularization term $\|X - G(X)\|_1$ was used to penalize making large changes at pixel level. This work is motivated by SimGAN, which is a model to reduce the gap between synthetic and real image distributions. However, unlike this prior work, this work focus on refining images with complex background and foreground objects, especially on road scene dataset.

Synthetic Images
Many efforts have explored using synthetic data for computer vision research, including gaze estimation [21], human pose estimation [16] and road scene [8, 15]. Gaidon *et al*. show that pre-training a deep neural network on synthetic data leads to improved performance on multi-object tracking. Unlike the above approaches, this work is to improve the realism of the complex synthetic images using unlabeled real data.

2. Methods

The proposed model is described in this section. Other similar variant models are also investigated for their applicability for this task. Limitations of these models are explained, and we propose a new architecture that can be used to refine complex images.

2.1. Architecture

The goal of this work is to learn mapping functions between two domains *X* and *Y* given synthetic images $x_i \in X$ and real images $y_j \in Y$. Figure 2(a) gives an overview of the proposed model: a synthetic image is refined using the generator network, $G_\theta : X \rightarrow Y$. The generator G_θ is trained to produce outputs that cannot be distinguished from real images by an adversarially trained discriminator, D_ϕ, which is trained to do as well as possible at detecting the generators fakes.

We adapt the encoder-decoder based generator architecture, G_θ, from Zhu *et al*. [22] to enable geometric change and also for efficient training. An input image of size 128x256x3 is first convolved with 7x7 filters that output 64 feature maps. This then followed by two 3x3 stride-2 convolution layers, seven ResNet blocks [6], and two deconvolution layers. Each ResNet block consists of two convolution layers containing 256 feature maps. All layers use BatchNorm and ReLU activation except for the output, which uses Tanh.

The discriminator network, D_ϕ, is similar to PatchGAN [7] with additional layers added for global loss information, as illustrated in Figure 2(c). The "local" path is a patch-level discriminator, which outputs $w \times h$ dimensional probability map of patches and aims to classify whether each patch in an image is real or fake. The $w \times h$ is the number of local patches in the image. This kind of architecture is also used in [17]. It provides many samples per image for learning the discriminator network. By limiting the discriminator's receptive field to local regions instead of the whole image, we can: (1) avoid drifting and prevent artifact. (2) apply the network to arbitrarily-sized images in a fully convolutional fashion. However, because the synthetic image might locally resemble real image, it is likely that the discriminator will be fooled. In other words, the generator G_θ could just be an identity map. To address this issue, I design the discriminator with an additional "global" path with much larger receptive field.

The implementation detail of the discriminator network is as follow: An input image of size 128x256x3 is passed through three 4x4 stride-2 convolution layers. The output

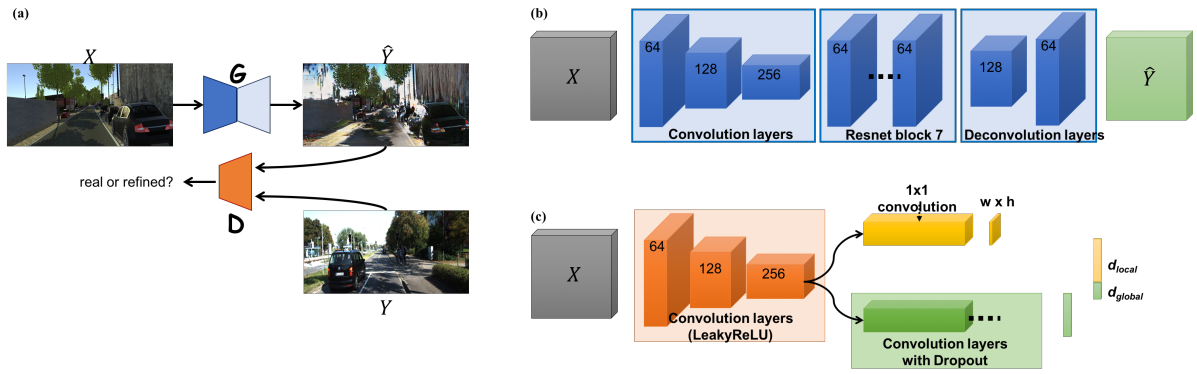

Figure 2. (a) Overview of our model. We refine the synthetic image with a generator network, $G : X \rightarrow Y$, that minimize the combination of adversarial loss and a "self-regularization" term (Equation 2). The discriminator D encourage G to translate X into outputs indistinguishable from domain Y. (b) Encoder-decoder based generator architecture to enable geometric change and also for efficient training. (c) Discriminator architecture: PatchGAN with additional global loss information. The dropout in the global path is needed to avoid oscillating behavior.

is passed to both "local" path and "global" path. The local path contains a Conv 1x1, stride-1 followed by a fully-connected layer, output a 512-dim vector d_{local}. The global path contains three 3x3 stride-2 convolution layers with dropout 0.4 (Conv-BatchNorm-LeakyReLU-Dropout). Is it then followed by two fully-connected layers and output a 16-dim vector d_{global}. Finally, we concatenate d_{local} and d_{global} into a single vector z. While training the generator network, we sum the cross-entropy loss values over z. The LeakyReLU activation is used in the discriminator for all layers. The dropout in the global path is needed to avoid oscillating behavior caused by the generator exploiting a weakness of the discriminator.

2.2. Objective

Adversarial Loss To add realism to the synthetic image, we need to bridge the gap between the distributions of synthetic and real images. An ideal generator will make discriminator impossible to classify a given image as real or refined. We denote the synthetic data distribution as $x \backsim p_{data}(x)$ and real data distribution as $y \backsim p_{data}(y)$ and express the objective as:

$$\mathcal{L}_{GAN}(G_\theta, D_\phi, X, Y) = \mathbb{E}_{x \backsim p_{data}(x)}[D_\phi(G_\theta(x))^2] + \mathbb{E}_{y \backsim p_{data}(y)}[(D_\phi(y) - 1)^2] \quad (1)$$

where G_θ tries to generate images $G_\theta(x)$ that looks similar to images from domain Y, while D_ϕ aims to distinguish between refined samples $G_\theta(x)$ and real samples y. The generator aims to minimize this objective against an adversary discriminator that tries to maximize it. Note that the discriminator network outputs a $w \times h$ probability map instead of a scalar, so that the adversarial loss function is the sum of the cross-entropy losses over the local patches.

Here, we apply the techniques from recent works [13] to stabilize the training process. The negative log likelihood objective is replaced by a least square loss, which is more stable during training and can generate higher quality results.

Self-Regularization To preserve the annotations of synthetic images, we follow Shrivastave *et al.*'s strategy [17] and complement the adversarial loss with a self-regularization loss that penalizes large changes between the synthetic and refined images.

$$\mathcal{L}_{reg}(G_\theta, D_\phi, X) = \mathbb{E}_{x \backsim p_{data}(x)}[(D_\phi(G_\theta(x)) - 1)^2] + \lambda \|G_\theta(x) - x\|_1 \quad (2)$$

where the first part of the cost adds realism to the synthetic images, which makes the discriminator fail to classify the refined images as synthetic, while the second part preserves the annotation information of the simulator. This self-regularization loss minimizes per-pixel difference between a feature transform of synthetic and refined images.

2.3. Training Detail

We learn the generator and discriminator parameters by minimizing $\mathcal{L}_{reg}$ and $\mathcal{L}_{GAN}$ alternately. We first pre-train both networks from scratch. Then, for each update of D_ϕ, we update G_θ twice. For training the discriminator network, each mini-batch consists of randomly sampled refined images $\hat{y}_i$ and real images y_j. The target labels for the cross-entropy loss layer are 0 for every $\hat{y}_i$; and 1 for every y_j.

In order to reduce model oscillation [4], we update the discriminator using a history of refined images [17] rather than only the ones produced by the latest generative networks. We keep an image buffer that store the 50 previously generated images.

Figure 3. Examples from the dataset used in our experiment. *(Left)* synthetic image from Virtual KITTI. *(Right)* real image from KITTI object detection.

For the road scene experiment, we set $\lambda = 10$ in Equation 2. We use Adam optimizer with a batch size of 8. All network are trained with learning rate of 0.0002 with momentum decay 0.5.

3. Experiments

We compare against previous approaches both qualitatively and quantitatively. For training and evaluating the model, we use three road scene datasets: (1) The KITTI "Object Detection Benchmark" [3] for real images; (2) The Virtual KITTI dataset [9] for synthetic images - samples shown in Figure 3; (3) The KITTI "Road/Lane Detection Benchmark" [2] for quantitative evaluation, for which we perform the segmentation of roads.

3.1. Quantitative Results

In this section, we first describe the segmentation model and metrics used in our experiments. We then report results regarding the differences between the virtual world and real world. Finally, we report our experiments on learning in refined world.

Road Segmentation Network Architecture We use the KittiSeg which is a sub-module in MultiNet [19] for doing road segmentation. The KittiSeg performs segmentation of roads by utilizing an Fully Convolutional Networks (FCNs) based model [12] that is trainable end-to-end. The output of the encoder is the 5th pooling layer which is called *pool5* in the VGG implementation [18]. The segmentation decoder follows the FCN architecture. First, a 1×1 convolutional layer is added after the encoded feature. This is followed by three transposed convolution layers to perform up-sampling and a 1×1 convolution layer.

Evaluation Metrics We evaluate the road segmentation performance using MaxF1 score (F_{max}) as in KITTI benchmark Suite [2]. The F-measure is derived from the precision and recall values for the pixel-based evaluation.

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

Since the method output confidence maps, the classification threshold τ is chosen to maximize the F, yielding F_{max}:

$$F_{max} = \underset{\tau}{\operatorname{argmax}}\, F \tag{4}$$

In addition, the average precision score as used in PASCAL VOC challenges [1] is also given for reference.

Figure 4. Different methods for generating refined images on Virtual KITTI dataset. From left to right: input (synthetic), SimGAN (refined), our work (refined).

Figure 5. Zooming in a portion of figure from top row of Figure 4. We can observe from the taillights of the car that this work has potential to produce visually more realistic feature.

Segmentation Performance All the network performance assessments are calculated by testing on real images from the KITTI "Road/Lane Detection Benchmark" data set. This dataset is very small, providing only 289 training images. We use 50 of them as the testing data for all the network. Training is conducted for 3300 iterations. The training for road segmentation is performed on the following four different sets and the performance comparison is reported in Table 1, Table 2 and Figure 6.

- **REAL** All KITTI "Road/Lane Detection Benchmark" training data set, besides the 50 for testing.
- **VIRTUAL** 2126 training images from Virtual KITTI dataset in the "morning" set.
- **SimGAN** The refined images produced from the *VIRTUAL* set above using SimGAN with $\lambda = 1.0$.
- **This Work** The refined images produced from the *VIRTUAL* set above using the model proposed in this work.

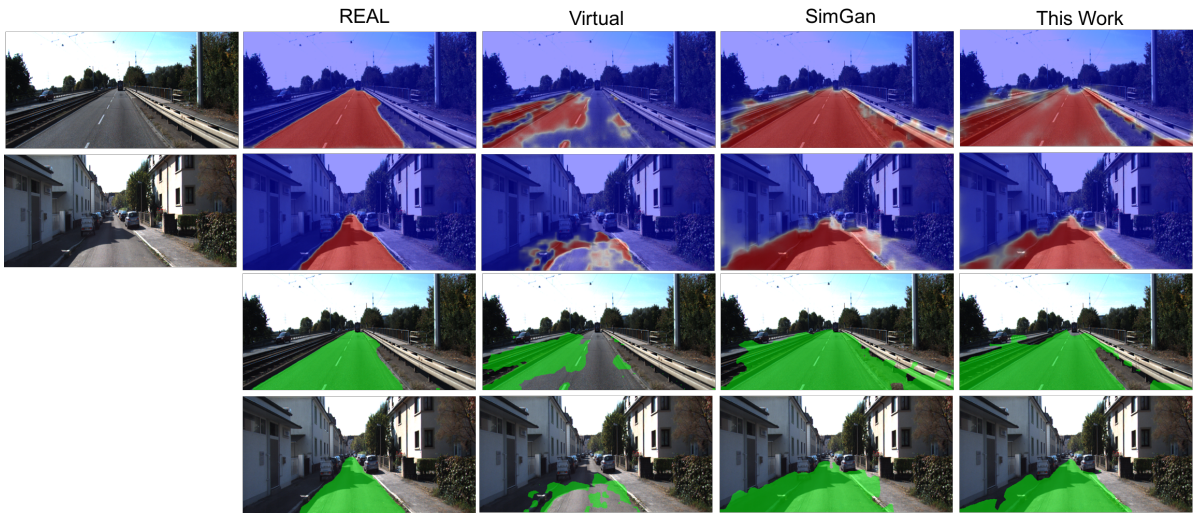

Figure 6. Visualization of the segmentation output using different training sources. Top rows: Soft segmentation output as red blue plot. The intensity of the plot reflects the confidence. Bottom rows hard class labels. From left to right: real image from KITTI dataset as testing input, trained using real images, trained using virtual images from Virtual KITTI dataset, trained using refined images from SimGAN, trained using refined images from this work.

Method	MaxF1
REAL	94.25
VIRTUAL	67.78
SimGAN	83.26
ThisWork	74.41

Table 1. Validation MaxF1 score of the road segmentation. Performance for the MultiNet network trained on synthetic, refined and real dataset are shown for comparison.

Method	Average Precision
REAL	91.89
VIRTUAL	54.39
SimGAN	83.13
ThisWork	76.77

Table 2. Validation Average Precision of the road segmentation.

As can be seen in Table 1 and Table 2, we observe a large improvement in performance from training on the refined images, at least 6.63% absolute percentage improvement. Although the refined results is unable to achieve compelling results with the real images, we believe that if we can generate more simulated $\rightarrow$ refined images, the performance will be comparable. Previous works, [8] and [9], have already proved that with larger number of only simulated images, they achieve superior performance to a network trained with only real images. [17] also observe a large improvement as the number of synthetic training examples is increased. Notice that although the MaxF1 score of this work is not better than SimGAN, our work can learn features that require geometric changes while SimGAN producing minimal changes to the input as discussed in section 3.2.

3.2. Qualitative Results

Figure 4 shows example output of SimGAN and this work on the Virtual KITTI road scene dataset. Notice that the refined image from SimGAN does not change largely since they use a fully convolutioinal neural network that operates on a pixel level rather than holistically modifying the image content. However, their method somehow is limited to shape change, providing only texture changes - which means that they can not handle tasks that require geometric changes. For example, we can observe from the taillights of the car in Figure 5 that this work has potential to produce visually more realistic feature.

4. Limitation and Discussion

Although our method can achieve some attracting results, the results are far from uniformly positive. The bad MaxF1 score is due to the translation degenerate the edge of the road, making it blurred. This might be caused by our generator architecture, which is an encoder-decoder based network, modifies the global information. Another cause might come from complex discriminator, leading it unable to learn well. Making it balanced between geometric transformation and texture-only transformation is an important task for future work. In addition, the pixel-wise L1 loss may be restrictive when the synthetic and real images have significant shift in the distribution, so we could try to re-

place the identity map with an alternative feature transform, such as learnt features from a mid layer from VGG [18].

References

[1] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. of Computer Vision*, 2010.

[2] J. Fritsch, T. Kuehnl, and A. Geiger. A new performance measure and evaluation benchmark for road detection algorithms. *ITSC*, 2013.

[3] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *CVPR*, 2012.

[4] I. Goodfellow, J. Pouget-Abadie, B. X. M. Mirza, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NIPS*, 2014.

[5] I. Goodfellow, J. Pouget-Abadie, B. X. M. Mirza, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative visual manipulation on the natural image manifold. *ECCV*, 2016.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[8] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016.

[9] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan. Virtual worlds as proxy for multi-object tracking analysis. *arXiv preprint arXiv:1605.06457*, 2016.

[10] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. *ICML*, 2017.

[11] M.-Y. Liu and O. Tuzel. Coupled generative adversar- ial networks. *NIPS*, 2016.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, 2015.

[13] X. Mao, Q. Li, H. Xie, R. Y. Lau, and Z. Wang. Multi-class generative adversarial networks with the l2 loss function. *arXiv preprint arXiv:1611.04076*, 2016.

[14] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[15] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. *CVPR*, 2016.

[16] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *PAMI*, 2013.

[17] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *CVPR*, 2017.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[19] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*, 2016.

[20] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graphics*, 2014.

[21] E. Wood, T. B. sand L. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesised images. *ACM Symposium on Eye Tracking Research and Applications*, 2016.

[22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017.