Synthetic Hong Kong Urban Image Generation

Alvin HARJANTO
HKUST
aharjanto@connect.ust.hk

Cheuk Hei CHONG
HKUST
chchongaa@connect.ust.hk

Kai Lok LAM
HKUST
kllamaf@connect.ust.hk

Urban planning is an important issue for a city, which could affect the development of the area. Visualization of designs would be one of the time-consuming tasks in the process. In our project, it is aimed to provide a solution of visualizing the photorealistic landscape picture from a draft design, in which designers could draw and state the types of objects by its corresponding colors. This solution is much less time-consuming than the current solutions.

1. Introduction

This project aims to generate photorealistic urban landscape images of Hong Kong from labelled segmentation maps, which will be useful in industries that need landscape visualization such as urban planning. To achieve this, we will conduct experiments with state-of-the-art deep learning models such as DeepLab and GauGAN. Datasets for training will be collected from readily available sources on the internet, and results will be evaluated both qualitatively and quantitatively.

2. Related Work

Image-to-image translation. This procedure is meant to transform an input image to a synthetic image or map an input image to the output image. Different models like pix2pixHD [1] and CycleGAN [2] are examples of this particular action. Pix2pixHD is one of the most common models that perform well alongside other typical conditional GANs. It will be explained more in next two parts.

Generative Adversarial Network. GAN [3] is a kind of unsupervised learning which can generate photorealistic or combined images based on the existing real images. It has been widely adopted in different applications including face generation, image-to-image translation, etc. More applications have risen recently in regards to utilizing GANs to generate realistic images from segmentation maps. However, it is seldom noticed that people are using GAN to conduct urban planning mock-up designs for showcasing. Therefore we chose to focus on generating urban landscape images for this project.

pix2pixHD. It is a very popular GAN architecture which is adaptable in different scenarios, including labels/segmentation map to photorealistic photos, adding colours to black & white photos and aerial photos to maps. [5] Similar to GauGAN, results are generally satisfactory, but it might not work well in some segmentation maps. As shown in Figure 1, when the input is spatially uniform, the instance normalization in pix2pixHD throws information away from the segmentation map. Also, with improvement of normalization in SPADE, it could perform better in generating photorealistic images in such scenarios, which made us use GauGAN instead.

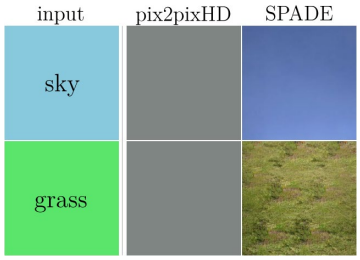

Figure 1 Output difference between pix2pixHD and SPADE (GauGAN)

3. Data

The data preparation can be divided into several phases.

3.1 Finding Image Source

The cityscapes dataset is captured from the street images of Germany, which contain a lot distinct features of European structures, differing from Hong Kong street images. To capture large amounts of street images of Hong Kong, we found several high-resolution car camera recordings on YouTube which provide more than 4000 images of different places in Hong Kong.

Figure 2 Sample Images from YouTube videos

3.2 Image Scraping

After finding the Hong Kong car camera recordings, image scraping is performed. OpenCV was used to extract frames from the video every 1.5 seconds, creating individual snapshots. The interval was used to reduce the

correlation between images, as all the videos consist of continuous shots.

3.2 Image Pre-processing

With the limited dataset image found on the Internet, data augmentation has been adopted, relevant operations including horizontal flipping, contrast and brightness randomization. Consider that the original resolution is relatively large, resulting in longer training time in the experiment. Therefore, downscaling is used, and the size of the training images consistently resized to 960 × 480. Image cropping has been conducted at the same to maintain the image scale in 2:1.

3.4 Filtering

As the image sources are from the frame captured in the videos, it is inevitable to obtain some frames with motion blurring. It is also observed that some frames include some wordings added by the video producers which might affect the image segmentation results. All those problematic frames are removed through the filtering process.

4. Method

Our approach is divided into two parts, segmentation generation and retaining of GauGAN models. Unlike Cityscapes dataset that already includes well-classified segmentation maps for every image, therefore before training on the GauGAN models, in the first stage, we have to utilize some existing models to generate the segmentation map on our own. We decided to adopt two existing models for segmentation: DeeplabV3 and FCHarDNet. Both models are able to classify different objects in our datasets, especially detection on humans, vehicles, buildings, roads, road signs, plants and the sky. As different models may generate different segmentation results and error is unavoidable, after the getting result, we will compare both models and choose the best optimal one in a quantitative and qualitative way.

It is then moved to the second stage, synthetic image generation by GauGAN models. With the generated image segmentation map in the previous step, original images and segmentation maps will be fed into GauGAN models to train. After training, we can use the testing image segmentation map to generate realistic images. The general structure of our network is shown in the figure below:

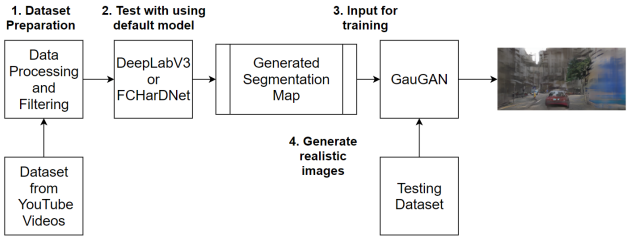

Figure 3 General Overview of Our Network

4.1 DeepLabV3+ and FCHarDNet

4.1.1 DeepLabV3+

DeepLab is a deep learning model for semantic image segmentation, which assigns semantic labels to pixels in the input image. [1] This could partition images into regions of types of objects. In DeepLabV3+, an encoder-decoder structure is used to extend DeepLabV3. It encodes the images with DCNN and apply atrous convolution. The decode module refines the segmentation results along the boundaries of objects.

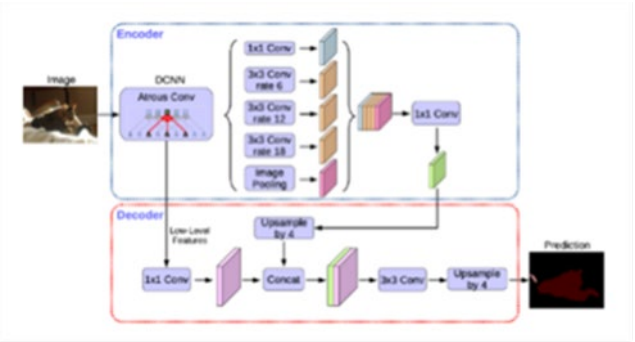

Figure 4 DeepLabV3+ architecture

4.1.2 Fully Convolutional Harmonic DenseNet (FCHarDNet)

FC-HarDNet is implemented based on a network architecture called Harmonic Dense Network. It uses a simple U-shaped encoder-decoder structure to produce loss for semantic segmentation.

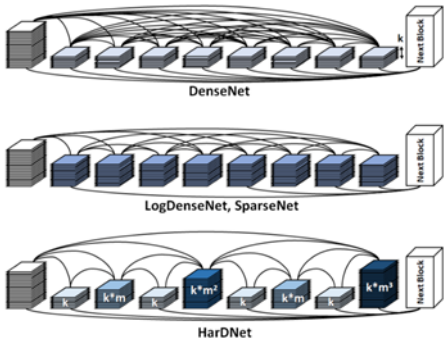

Figure 5 comparison of DenseNet, LogDenseNet and HarDNet

The HarDNet connects the layers k and layers k-2^n. Each layer is a 3x3 convolution.

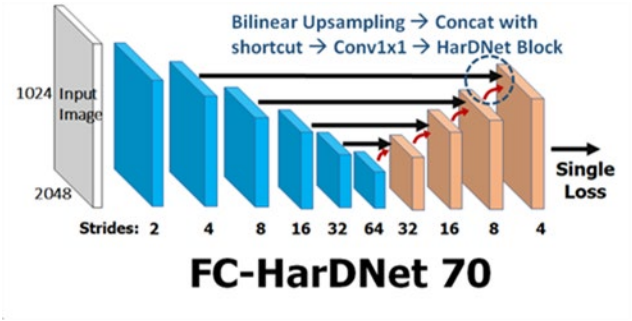

Figure 6 Architecture of FCH

4.2 GauGAN

GauGAN is a GAN network developed by Nvidia which creates photorealistic images from segmentation maps, which are labelled sketches that depict the layout of a scene. Talking about the architecture of GauGAN, it contains a generator and a discriminator. The discriminator performs classification of each pixel in an image and its structure is mainly based on pix2pixHD. The segmentation map and the image will be taken as the concatenation as the input. Then it will pass through several convolutional layers and a instance norm layer.

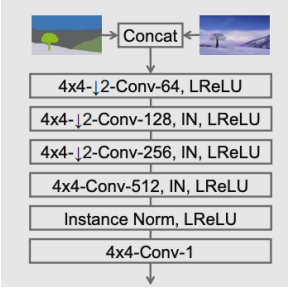

Figure 7 Illustration on Discriminator

For the generator, it is a convolutional decoder based on SPADE blocks. The design involves resizing layer, the 3x3-Conv-k layers, as well as the batch norm layer.

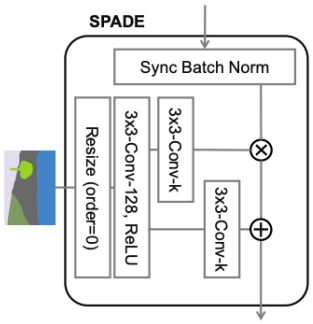

Figure 8 Illustration on Generator

For SPADE normalization layer, batch norm is modified based on the pix2pixHD one. Recall the problem in pix2pixHD that loss of semantic information is contributed to the unconditional normalization, in SPADE, learning multiple sets of parameters for each pixel is adopted instead of learning single set of parameters.

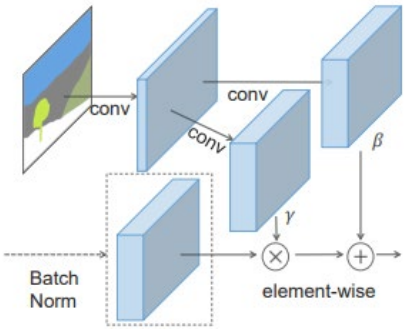

Figure 9 Illustration on SPADE

5. Experiment

In this section, we will first compare and choose the optimal model for segmentation map generation, then proceed to analyze and evaluate the GauGAN model for final synthetic image generation.

5.1. Experiment Environment

All the process, including image segmentation map generation and synthetic image generation are operated in Google Colab Pro environment, configured with a Nvidia T4/P100 GPU, 16GB DDR6/ 12GB HBM2 memory capacity. It is needed to emphasize that GauGAN training requires tremendous computational power, which our hardware is incomparable with the laboratory environment in SPADE paper using 8 32GB V100 GPUs at the same time.

5.2. Segmentation Evaluation

As discussed above, we conduct experiments on both segmentation models, namely FCHarDNet and DeepLabV3 using several quantitative and qualitative methods. These 2 models are chosen as they claim to achieve an mIoU accuracy of 76.0% and 83.1% [4] on the Cityscapes dataset, respectively.

Figure 10 Sample test image from custom dataset

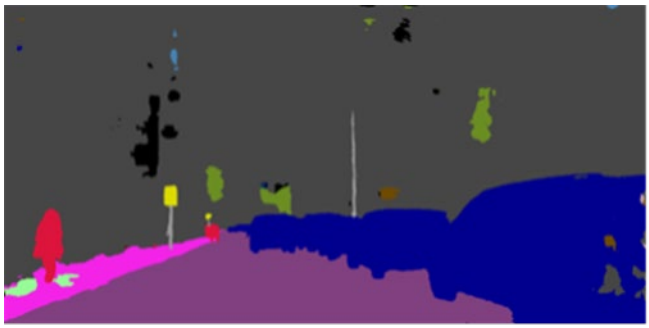

Figure 11 Segmentation map produced by DeepLabV3

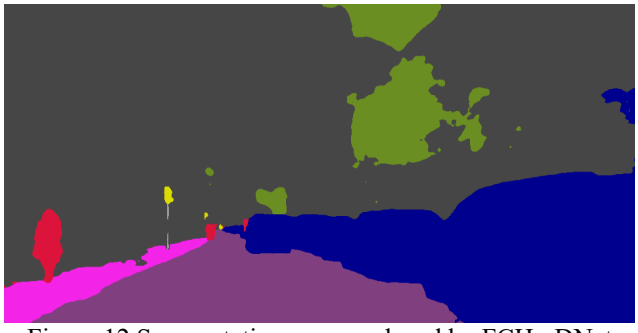

Figure 12 Segmentation map produced by FCHarDNet

	mIoU	**F1-score**
DeepLabV3	0.253	0.284
FCHarDNet	0.445	0.513

Table 1: Quantitative analysis on segmentation model performance

As visible through the table, the score achieved on both models are inferior and deviate far from the ideal accuracy. This is expected, as both models were trained on the Cityscapes dataset, which comprises Germany cityscapes images. Hong Kong's cityscapes are significantly different, having far more complex building structures and elements that are not even contained as part of the segmentation labels, e.g., scaffolding. An ideal approach would be to train both models using a custom dataset on Hong Kong, however this is infeasible as generating ground truth segmentation maps will be very costly, require manual work and lie beyond the scope of this project.

Qualitatively, it is visible that DeepLabV3 generates more detailed segmentations and tends to be pessimistic during classification, in turn generating less false positives, e.g., successfully separating cars from their shadows. However, this leads the model to generate more false negatives as well (shown by black spots on the buildings).

Meanwhile, FCHarDNet tends to be very optimistic during classification, causing a lot of false positives to be generated. While DeepLabV3 performs significantly better on Cityscapes, it is important to note that the model is far more complex, hence being more prone to overfitting towards the dataset it was trained on, which seems to be the case here. Additionally, DeepLabV3 also outputs 32 classes, compared to 20 for FCHarDNet. Based on these analyses, we decided to go forth with FCHarDNet, as due to limited dataset resource, we favor a simpler model. It also proves to produce more promising results for our custom dataset.

5.3. GauGAN Evaluation

Following segmentation map generation, we utilized the SPADE Network by Nvidia to generate synthetic images. The model was trained purely on the custom dataset of 4,203 images, with specifications as follows:

Specification	Value
Epochs	100
Batch size	1
Learning rate	0.0002
Initialization	Xavier
Optimizer	Adam

Tables 2 : GauGAN model training specifications

For evaluation purposes, we create a snapshot of the model losses every 100 iterations. The diagrams below display the loss history over snapshots.

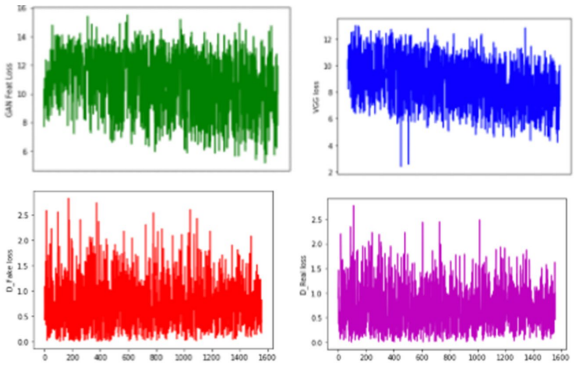

Figure 13 Loss history diagrams

From the graphs, we can see that although the losses oscillate significantly, all losses have a consistent decreasing trend. As an example, the Generator loss (Figure 4: top-left) improves to a loss of 10 from an initial mean loss of 12. The discriminator losses (Figure 4: bottom) also show a constant decline on maximum values per snapshot. This consistent improvement indicates that the GauGAN was constantly learning and slowly converged to lower losses, despite the flawed dataset.

After 70 epochs, we realized that due to the relatively small dataset, the model was starting to overfit towards specific patterns on the training images. Therefore, we decided to add more images to our dataset in the hope of reducing overfitting and continuing training. Below is a sample result after 100 epochs.

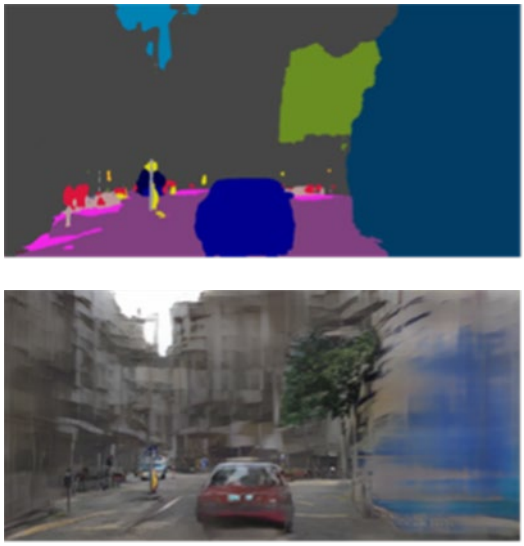

Figure 14 Sample results of GauGAN

Quantitatively, we will evaluate the model using 2 metrics, mIoU test and FID score via random sampling from the dataset. We will also compare these metrics with previous models. Since we are using a custom dataset of cityscapes, we will use models trained on Cityscapes as a benchmark. [3]

	Cityscapes		Custom	
	mIoU	FID	mIoU	FID
CRN	52.4	104.7	-	-
SIMS	47.2	49.7	-	-
pix2pixHD	58.3	95.0	-	-
Ours	-	-	25.76	67.19

Table 3: Quantitative analysis of GauGAN results

Note that the comparisons are not perfect, as there are significant differences between the Cityscapes dataset and our customs dataset. As visible through the table, the model evaluated on FID performs comparably well compared to its predecessors. However, the mIoU test reveals poor results. This might be attributed to the segmentation models we were using which overfits the Cityscapes dataset, hence the evaluation cannot be fully accurate.

Qualitatively, we can see that Figure 5 shows promising results after 100 epochs, successfully generating accurate representations of each segmentation element. However, the output image lacks detail, especially on elements that appear less common in our dataset, such as buses (denoted by pale blue). Buildings (denoted by gray) are also not well-defined, as Hong Kong cityscapes possess extremely complicated structures.

Since the losses were still converging, it is possible that further training and additional datasets will improve the results. However, due to limited data and computing resources, we were unable to achieve the ideal results.

Figure 15 GauGAN generated results, random snapshots over training

6. Conclusion

In this project, we have tried to retrain GauGAN with our custom-made datasets, which are images captured from car camera videos in Hong Kong, to synthesize a photorealistic

image from simple segmentation images. However, our results are still room for improvement due to computing resources. During the retraining, it is found that there are a lot of hyperparameters that need to be optimized and each iteration and epoch could take days and weeks. With only Google Colaboratory Pro, it is much harder and time-consuming to optimize the training and prevent overfitting.

It is learnt that how to do preparing custom datasets if the one needed does not exist. In the first part of the project, we have tried to find a dataset with local street images and corresponding ground truth segmentation maps. Nevertheless, the dataset needed could not be found. As a result, we have tried to make our own datasets with several sources and segmentation computation models. In our experiments of segmentation computation, both DeepLab and FCHarDNet could generate segmentation maps with correct labels in most of the cases. However, the generated datasets behave differently with GauGAN. FCHarDNet was chosen in the project for datasets preparation due to better fitting in GauGAN.

Other than datasets preparation, it is also learnt on working with a model with custom datasets even though the results of our work are not ideal. The work is more complicated and demanding than expected.

Some extensions to this project could be made for new applications. To synthesize photorealistic street images of a specific, the model has to be retrained and optimized with more epochs. This could be done by people with more computation resources and time. The better well-trained model may work well with the synthesis of images for other city with similar features. The final well-trained model could be used for urban planning as the initial goal of this project. A software could be built along with this model to provide instant realistic simulation from designers' and engineers' sketches. The changes on the sketches on canva could be reflected on the synthesized realistic images on render. The results of the construction could be visualized by some simple sketches. This could ease the traditional way of urban planning by reducing the workload of building CAD and simulation.

References

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," Google Inc, 2018.

[2] Ping Chao, Chao-Yang Kao, Yu-Shan Ruan, Chien-Hsiang Huang, Youn-Long Lin, "HarDNet: A Low Memory Traffic Network," National Tsing Hua University, University of Michigan, 2019.

[3] T. Park, M.-Y. Liu, T.-C. Wang and J.-Y. Zhu, "Semantic Image Synthesis with Spatially-Adaptive Normalization," in *IEEE Conference on Computer Vision on Pattern Recognition*, 2019.

[4] L.-C. Chen, G. Papandreou, F. Schroff and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," in *IEEE Conference of Computer Vision and Pattern Recognition*, 2017.

[5] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," in *IEEE Conference of Computer Vision and Pattern Recognition*, 2018.

[6] J. P.-A. M. M. B. X. D. W.-F. S. O. A. C. Y. B. Ian J. Goodfellow, "Generative Adversarial Networks," 2014.

[7] T. P. P. I. A. A. E. Jun-Yan Zhu, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," 2017.