International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 12378-12383

Customized Semantic Segmentation by using Virtual World Data

JueSeok Kang[1], DongKeun Kim[2] and EunJoo Rhee[3]

[1,2]M&S Center 1, SIMNET, Daejeon 34127, Korea.
[3]Professor, Department of Computer and Engineering, Hanbat National University, Daejeon 34158, Korea.

[1,2,3]Orcid: 0000-0001-8454-4248, 0000-0001-6093-126X,
[3]Corresponding Author, [3]0000-0002-3783-2359

Abstract

Convolutional based networks are general methods for computer vision tasks. Despite their remarkable performance, they require huge amounts of datasets and sometimes customized datasets must be used for specific purposes. However, customized datasets require a great deal of time for generating and labeling. To solve this problem, we propose a virtual world that is specialized for semantic segmentation of a driving environment analysis. We use these data to train a FCN(Fully Convolutional Network) network and show competitive results. We also assess the applicability of this model to a real world driving situation.

INTRODUCTION

Convolutional neural networks (CNNs) are widely used models for computer vision tasks such as image classification, object localization, and object detection. CNNs have many practical applications such as autonomous driving, object tracking, and action recognition [1].

Although the performance of these tasks using convolutional neural networks is improving, a large amount of labeled data is required for learning to achieve required performance.

There are several available public datasets such as ImageNet datasets and PASCAL VOC datasets. However, there are some limitations to using these datasets when we want to detect or classify specific objects that are not contained within them. In these cases, we need to create customized datasets and this requires time and labor for manual labeling procedures.

Thus, in this paper, we construct a virtual environment to create a customized dataset for semantic segmentation, specifically for a driving environment analysis. By comparing the training results using the data obtained in the virtual environment, the training results using an actual car black box image, and the training results using both virtual and car black box data, we determine that it is possible to use virtual world data in a real world case, and we can thus shorten the process of dataset labeling.

RELATED WORK

Semantic segmentation is a problem that assigns each pixel in the image to each object class. A Fully Convolutional Network (FCN) uses a deconvolution operation to predict the object class of each pixel, and showed high performance in benchmark data such as PASCAL VOC [2].

In the paper “Learning Deconvolution Network for Semantic Segmentation” [3], they use VGGNet[4] as a front part, and use several deconvolution layers and unpooling layers as a rear part to address the disadvantage of FCN, which is hard to delineate because of its simple deconvolution layer structure.

Virtual world construction for data acquisition and training is widely conducted. In addition, the applicability of virtual world training results to the real world is also tested. Erik Bochinski et al. constructed a virtual world for object tracking and confirmed that it is applicable to object tracking in a virtual world[5].

In recent deep learning studies, efforts have been made to verify the performance of models using datasets such as PASCAL VOC, which provides a large amount of images and labeled data constructed in the past or a virtual environment such as GTA5.Artur.Fliipowicz et al. conducted a study on autonomous driving in a virtual world using a commercial game GTA5, and showed that it can perform human-level game driving [6].

However, there is a problem that it is difficult to use the pre-trained model for specific purposes, and the previous studies have been carried out using data acquired in the real world where a lot of labeling work is required [2, 3]. Moreover, studies using the virtual world only have focused on improving performance in the virtual world [6], or have been limited to object detection [5].

Upon this background, in this study we combine data obtained from the virtual world and the real world to reduce resources required for labeling and guaranteed the required performance for semantic segmentation in a real world application.

SYSTEM MODEL AND METHODS

First, a 3D-based virtual world that training can be performed in a virtual world is constructed. By using the virtual world, a virtual camera image and its labeling data for semantic segmentation are obtained to minimize the time and labor required to prepare for supervised learning.

Next, a car black box image is labeled by a user manually to perform supervised learning in a conventional manner.

Based on these processes, we propose an efficient method to shorten resources for labeling and training a large amount of data to obtain high performance.

A. Virtual world data generation and labeling

To generate customized semantic segmentation datasets, we use the Unity3D engine. This is an engine for game production, and many games have been developed through the Unity3D engine. It supports the functions required to configure the virtual world. We therefore use it in our research and generate labeled data by the following flow.

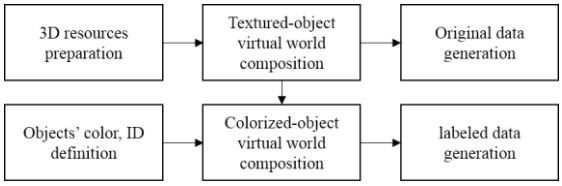

Figure 1: Virtual data generation process

In this research, we constructed a 3D virtual world and obtained labeled data according to **Fig. 1**. First, for the purpose of constructing the virtual world, we use prebuilt 3D resources for generating roads, buildings, cars, people, and so on. Next, each terrain, geographic feature, and object are divided into 48 classes, and the color corresponding to each class is defined. Through these processes, a textured-object virtual world for a driving simulation was constructed using these resources and class definitions.

For the purpose of convenient labeling, the textured-object virtual world is copied and each object's texture is replaced with each object's allocated color. Through these processes, a colorized-object virtual world is formed and each object has its own color.

Finally, the textured-object virtual world and the colorized-object virtual world are formed. After construction of the virtual worlds, we create a path by taking arbitrary coordinates in the virtual world and set the camera to follow the path. As the camera moves, two virtual worlds are captured according to the movement of the camera. The textured-object virtual world generates original image data, and the colorized-object virtual world generates labeled data.

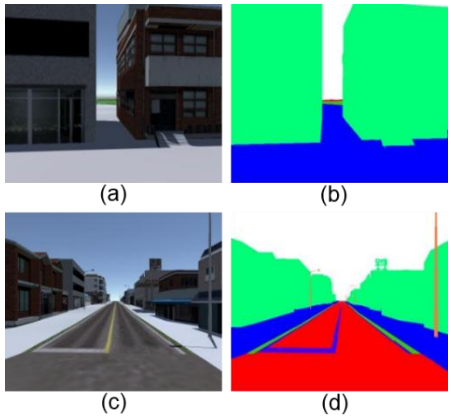

Figure 2: Examples of captured image from the textured-object virtual world and the colorized-object virtual world.
(a) Textured-object virtual world image for buildings.
(b) Colorized-object virtual world image for buildings
(c) Textured-object virtual world image for buildings and roads.
(d) Colorized-object virtual world image for image (c).

Fig. 2 shows examples of captured images from the textured-object virtual world and the colorized-object virtual world. **Figs. 2 (a)** and **(c)** are captured images from the textured-object virtual world, and **Figs. 2 (b)** and **(d)** show an example of a labeling image rendered by mapping colors assigned to textures. As shown in the figure, the labeling images used in the supervised learning have very clear and separated boundaries with no noise.

B. Real world data generation and labeling

In this paper, car black box images are used as real world data. Specifically, 36 car black box images are labeled and used as raw data for the real world experiment.

Data augmentation techniques widely used for enhancing the performance of Convolutional Neural Networks (CNNs) are applied to car black box images to extend the insufficient number of images. As shown in **Fig. 3**, 640 x 480 size images are cropped with five 480 x 360 size images as top-left, top-right, bottom-left, bottom-right, and center. Each cropped image is also horizontally flipped, and thus one original image was augmented to 10 images.

A total of 360 labeled images were obtained through this process and additional training was performed by adding to the existing images from the virtual world.

Figure 3: Data Augmentation Method Description. Each black box rectangle represents a raw image and colored rectangles represent augmented images.

C. Class definition

In this paper, the classification of virtual or real world camera images is systemized using a separate classification system, as shown in **Table 1** with own criteria.

Also, in order to develop an intelligent combat simulator for driving and shooting in the future, we defined 48 classes that are related to that purpose.

Table 1: Object Label and Color Definition

Category	Color (R,G,B)	Category	Color (R,G,B)
Asphalt	(255, 0, 0)	Unpaved road	(0, 255, 0)
Cement road	(0, 0, 255)	Forest	(255, 255, 0)
Stone pit	(0, 255, 255)	Mountain	(255, 0, 255)
Sidewalk	(255, 60, 0)	Farm	(255, 0, 60)
Grassland	(60, 200, 0)	Gravelly field	(0, 255, 60)
Sand	(0, 60, 255)	Boundary stone	(60, 0, 255)
Pedestrian overpass	(255, 120, 0)	Tunnel	(255, 0, 120)
Crosswalk	(120, 0, 255)	Building	(0, 120, 255)
Tree	(120, 0, 255)	Telephone pole	(0, 120, 255)
Street lamp	(255, 120, 60)	Traffic light	(255, 60, 120)
Traffic sign	(60, 120, 255)	Soldier	(120, 60, 255)
Man	(60, 255, 120)	Women	(120, 255, 60)
Child	(120, 60, 0)	Dog	(120, 0, 60)
Cat	(60, 120, 0)	Chicken	(0, 120, 60)
Truck	(0, 60, 120)	Bus	(60, 0, 120)
Car	(180, 120, 0)	Bike	(180, 0, 120)
Van	(120, 180, 0)	Tank	(0, 180, 120)
Armoured vehicle	(0, 120, 180)	Trailer	(120, 0, 180)
Barricade	(200, 170, 40)	Plastic cone	(200, 40, 170)
Chopper	(170, 200, 40)	Airliner	(40, 200, 170)
A fighter	(40, 170, 200)	Bird	(170, 40, 200)
Boat	(255, 150, 150)	Ship	(150, 255, 150)
Sign	(150, 150, 255)	Information	(0, 0, 0)
Sky	(255, 255, 255)	Water	(120, 120, 120)

D. Mozel description

The model used in this paper is FCN (Fully Convolutional Network) [2] and consists of two parts, the forward convolutional network and the deconvolutional network.

The forward convolutional network is based on the structure of VGGNet[4] and consists of 16 convolutional layer and 5 pooling layer. **Table 2** shows the detailed structure of the forward convolutional network. Layer represents the name of each layer, dimension represents the number of filters x width of filter x height of filter, and # layer means how many layers having the same structure are repeated. In addition, to avoid an over-fitting issue due to the increase in the number of layers, we add a dropout layer with a rate of 0.5 following Conv 6 and Conv 7. The forward convolutional network extracts features from images.

Table 2: The Structure of Forward Convolutional Network

Layer	Dimension	# layer
Pad 0	100 x 100	1
Conv1	64 x 3 x 3	2
Pool 1	2 x 2	1
Conv 2	128 x 3 x 3	2
Pool 2	2 x 2	1
Conv 3	256 x 3 x 3	3
Pool 3	2 x 2	1
Conv 4	512 x 3 x 3	3
Pool 4	2 x 2	1
Conv 5	512 x 3 x 3	3
Pool 5	2 x 2	1
Conv 6	4096 x 7 x 7	1
Conv 7	4096 x 1 x 1	1
Score fr	48 x 1 x 1	1

The rear part of the model, the deconvolutional network also followed FCN. In this paper, FCN-16s and FCN-8s model are used, and they are distinguished by the stride difference.

Table 3 and **Table 4** show the detailed structure of the deconvolutional network. Layer represents the name of each layer, dimension represents the number of filters x width of filter x height of filter, and stride means the upsampling factor.

For FCN-8s, the output of the score fr layer is doubled and fused with the output of pool 4 layer. Next, this value is doubled and fused with the output of pool 3 layer, and finally deconvolution operation of stride 8 restores the original image size, which was reduced through the forward convolutional network operation.

Table 3: The Structure of FCN-8s Deconvolutional Network

Layer	Dimension	Stride
Deconv 1	48 x 4 x 4	2
Fuse pool 4	-	-
Deconv 2	48 x 4 x 4	2
Fuse pool 3	-	-
Deconv 3	48 x 48 x 48	8
Softmax	-	-

For FCN-16s, the output of score fr layer is doubled and fused with the output of pool 4 layer, as in FCN-8s. The deconvolution operation of stride 16then restores to 640 x 480 size, which is the original image size.

Table 4: The Structure of FCN-16s Deconvolutional Network

Layer	Dimension	Stride
Deconv 1	48 x 4 x 4	2
Fuse pool 4	-	-
Deconv 2	48 x 32 x 32	16
Softmax	-	-

EXPERIMENTS AND RESULTS

To assess the performance of the proposed method, we carried out two classification experiments. Experiments are performed on an Intel Core i7-7700 CPU @ 3.60GHz, two NVIDIA GeForce GTX 1080 Ti Graphic card, 8GB RAM. At all times, we use 640 x 480 size image as the input image size.

For training using virtual world data, 801 images are trained for 36 epochs, and 89 images are tested. Pixel accuracy is used as a metric for semantic segmentation.

For training using real world data, 324 images are trained for 12 epochs, and 36 images are tested. Also, pixel accuracy is used as a metric for semantic segmentation.

First, the model trained using virtual world data is tested to check the performance. Next, the model trained using only the virtual world, the model trained using only the real world, and the model trained using both the virtual and real worlds are compared to test their applicability to the real world.

A.Virtual world results

Table 5 shows the semantic segmentation results of the virtual environment for the FCN-8s model and the FCN-16s model. As we expected, the results show that the FCN-8s model performs better than the FCN-16s model.

In **Table 5**, Pixel accuracy (Train) is the accuracy measured during training and Pixel accuracy (Test) is the accuracy measured after training by using different data that are not used in the training process.

Table 5: The Virtual World Results

	Pixel accuracy (Train)	Pixel accuracy (Test)
FCN-8s	98.90	98.72
FCN-16s	98.73	98.31

Detailed examples are shown in **Fig 4**. Contrary to well classified large classes such as roads, terrains, and buildings, fine structures such as trees and telephone poles are not clearly delineated due to a lack of training examples.

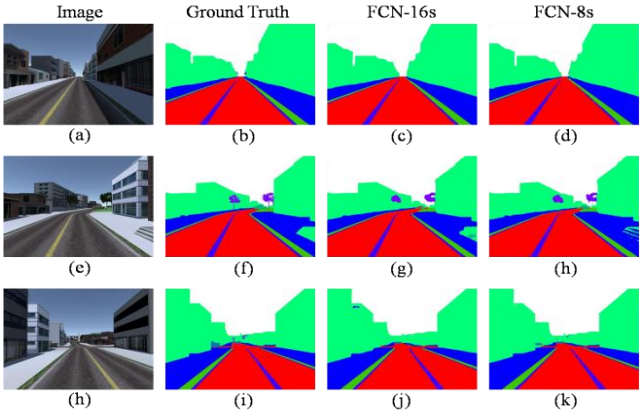

Figure 4: Visual representations of semantic segmentation. The left column shows the original images. The second column shows the ground truth for semantic segmentation. The third column shows the output of FCN-8s. The fourth column shows the output of FCN-16s.

B. Real world results

Considering **Table 5**, models trained using virtual world data appear to be sufficient for semantic segmentation of images obtained from the virtual world. If proposed model can be applied to images obtained from the real world, data preparation resources for supervised learning can be remarkably shortened.

For the experiment, real world images are labeled according to 48 pre-defined classes as virtual world images, as shown in Fig. 5.

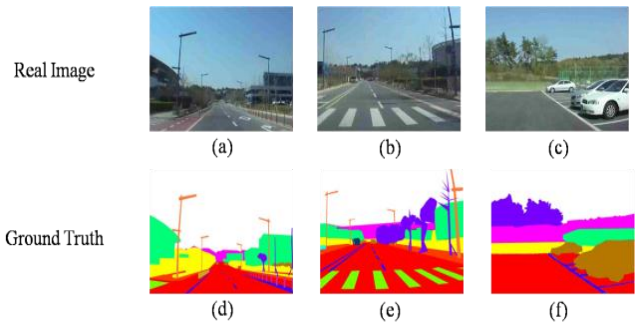

Figure 5: Examples of real world images and the labeled ground truth. (a), (b), (c) are car black box images. (d), (e), (f) are the ground truth for semantic segmentation.

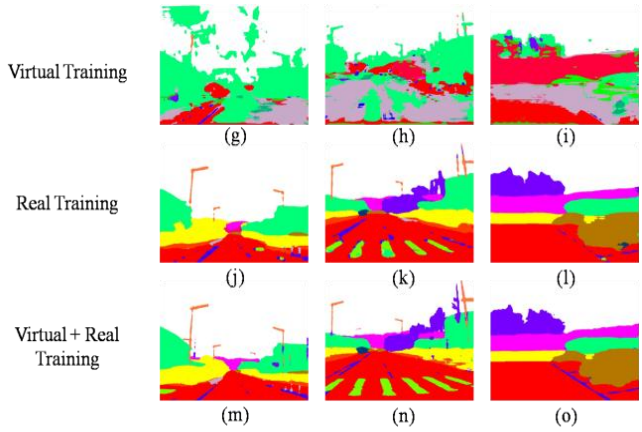

Figure 6: Visualizations of semantic segmentation outputs. The first row shows real image segmentation results for the model trained in the virtual environment (g, h, i). The second row shows results for the model trained by using only real images (j, k, l). The third row shows results for the model trained by using both virtual and real world data.

In the first experiment, the model trained using the virtual world images is used to segment real world images. **Figs. 6(g)**, **(h),** and **(i)** show the semantic segmentation results of the experiment for the images in **Fig. 5**. The results are very inaccurate although large boundaries are divided. The pixel accuracy of the first experiment is 55.52%, as shown in the first row of **Table 6**. This shows that it is difficult to apply the model trained using only virtual world data to a real world image directly.

In the second experiment, the model trained using the real world images is used to segment real world images. The results are shown in **Figs. 6 (j)**, **(k)**,and **(l)**. The pixel accuracy of the second experiment is 90.50%, as shown in **Table 6**.

In the last experiment, the pre-trained model using the virtual world data is re-trained using the real world data. The results are shown in **Figs. 6 (m)**, **(n)**, and **(o)**. The pixel accuracy of the last experiment is 93.90%, as shown in **Table 6**. Comparison with the second experiment showed that pre-training using virtual world data provided a 3.4% enhancement in the performance of semantic segmentation.

Table 6: The results of a real world experiment

Training method	Pixel accuracy (Train)	Pixel accuracy (Test)
Virtual training	-	55.52%
Real image training	89.65%	90.50%
Virtual &real image training	94.53%	93.90%

The results show that pixel accuracy can be increased by using the pre-trained virtual world model.

However, as shown in **Fig. 6**, the semantic segmentation results differ in accuracy from class to class. This might be due to an insufficient number of images that contained small objects during the virtual world and real world training. In addition, it is known that the FCN model used in this paper has limitations in delineating small size objects because of a pre-defined filter size that is not small enough to segment small objects [3].

In this regard, we will consider generating and labeling an additional dataset or improving the performance of the model we used for further studies.

CONCLUSIONS

In this paper, we adopt the virtual world as a method to reduce the time and labors required to perform labeling for supervised learning, and generate a large amount of labeled data in a short time. Experiments have shown that the virtual world can be used to improve the accuracy of semantic segmentation results despite a lack of real world labeled images.

By using the proposed approach, the pixel accuracy of the model produces similar results to the existing model for semantic segmentation, and confirms that virtual world training can be applied to a real world situation.

In this paper, we showed that CNN based models can be used to replace artificial intelligence in a virtual environment. Furthermore, we found that when the acquisition of real world data is limited, a combination of data obtained from the virtual world and a few labeled data from the real world can compensate for insufficient real labeled images.

REFERENCES

[1] A.Bhandare, M.Bhide, P. Gokhale and R.Chandavarkar, 2016, "Applications of Convolutional Neural Networks", *International Journal of Computer Science and Information Technologies* (IJCSIT), Vol.7 (5), pp. 2206-2215.

[2] J. Long, E.Shelhamer and T. Darrel, 2015, "Fully Convolutional Networks for Semantic Segmentation", *Computer Vision and Pattern Recognition* (CVPR), pp. 3421-3440.

[3] H. W. Noh, S. H. Hong and B. H. Han, 2015,"Learning Deconvolution Network for Semantic Segmentation", *International Conference on Computer Vision*, pp. 1520-1528.

[4] K.Simonyan and A. Zisserman,2015, "Very Deep Convolutional Networks for Large-Scale Image

Recognition", *International Conference on Learning Representations*,pp.1-14.

[5] E.Bochinski, V.Eiselein and T.Sikora, 2016, "Training a Convolutional Neural Network for Multi-Class Object Detection Using Solely Virtual World Data", *Advanced Video and Signal Based Surveillance*, pp. 278-285.

[6] A.Filipowiez, J. Liu andA.Komhauser, 2016 "Using Virtual Worlds, Specifically GTA5, to Learn Distance to Stop Signs", *TRB 96th Annual Meeting Compendium of Papers*,pp. 1-27.