Specific Area Style Transfer on Real-Time Video

Geun Tae Kim, Hyunmin Kim, Hyung-Hwa Ko

***Abstract:** Since deep learning applications in object recognition, object detection, segmentation, and image generation are needed increasingly, related research has been actively conducted. In this paper, using segmentation and style transfer together, a method of producing desired images in the desired area in real-time video is proposed. Two deep neural networks were used to enable as possible as in real-time with the trade-off relationship between speed and accuracy. Modified BiSeNet for segmentation and CycleGAN for style transfer were processed on a desktop PC equipped with two RTX-2080-T_i GPU boards. This enables real-time processing over SD video in decent level. We obtained good results in subjective quality to segment Road area in city street video and change into the Grass style at no less than 6(fps).*

***Keywords** : Deep Learning, GAN, Semantic Segmentation, Style Transfer*

I. INTRODUCTION

Recently, various research using deep learning has been conducted. Deep learning is widely used in real life, and the image processing is also increasingly needed in various fields. Examples of image processing and deep learning are surveillance, object detection, object tracking, and style transfer [1]. Among these applications, the object detection, which is currently used in autonomous driving technology, requires high accuracy with high speed. In the case of object detection, it is difficult to determine the exact position because the object is detected by the box rather than the exact position. Segmentation is used to accurately determine the position of an object. Segmentation is a method that combines classification and localization. All judgments are performed in pixel units in segmentation. However, it takes a long processing time [2-4]. Style transfer is one of the ways to create a new video. For example, taking Monet's style and applying it to the landscape around us, we make the landscape possess Monet's style. The reason using style transfer method is to give experience different environment to users. Some places for experiencing augmented reality or virtual reality are emerging. However, considering the limited content and time to develop the content, the supply is very limited compared to the demand. Therefore, if you use style transfer for content, you can create more content that you want. In this paper, after segmenting a specific area, we propose an algorithm that converts the road area into a desired style, such as Grass style. The purpose of the research is to implement in real-time video with subjectively satisfactory results using the trade-off between speed and accuracy. The paper is organized as follows. In section 2, related works are introduced, and algorithms for segmentation and style transfer of specific area are proposed in section 3. After describing the training methods in section 4, experimental results and discussion are written in section 5, and we concluded in section 6.

Manuscript received on February 09, 2021.
Revised Manuscript received on March 03, 2021.
Manuscript published on March 30, 2021.
* Correspondence Author

Hyung-Hwa Ko*, Dept. of Electronics and Communications Eng., Kwangwoon Univ., Seoul, Korea. Email: hhkoh@kw.ac.kr

GeunTae Kim, ABH Inc., Ulsan, Republic of Korea Email: kjss0521@naver.com

Hyunmin Kim, Telecons Inc., Seoul, Republic of Korea Email: hmkim@telecons.co.kr

II. RELATED WORKS

A. *Style transfer [1]*

Style transfer refers to a method of changing the style to be like the desired style when the content images are given, while maintaining shape of the content. Methods that have been announced so far for style transfer are using a Convolutional Neural Network (CNN) and Generative Adversarial Network (GAN). Regarding to CNN style transfer, it requires two input images consisting of a style image and a content image. Each style and content image comes out from each layer of CNN. In the case of a style image, the global array information of the style scene is deleted, and an image matching the style of a specific image with an increasing scale is created as shown in Fig. 1. In the case of the content image, it was almost like the original in the lower layer, and the image is preserved in the higher layer, but detailed pixel information is lost [1]. Because it is based on a pre-trained model, style transfer is possible even with two images. When converting a new content image, re-training is not necessary because only the input to the pre-trained image conversion network is changed, but the style image is limited to artistic images.

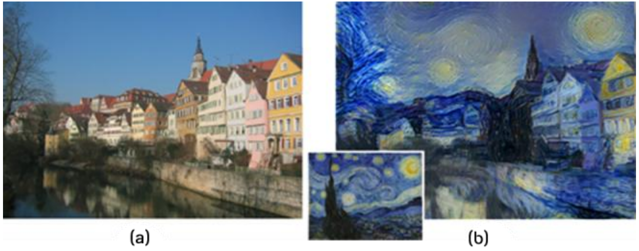

Fig. 1. (a) Content Image, (b) Styled Image.[1]

Generative Adversarial Network (GAN) [5] that consists of a generative network and a discriminative network is a network that improves performance by training adversarial with each other. While the generative network generates the good-looking fake image to make the discriminator network misjudge it as a real image, the discriminative network itself learns to increase the probability that the image generated from the generative network is not real.

Cycle GAN which means Cycle-Consistent Adversarial Networks [6] is the control GAN.

The shape of the model constitutes the structure of drawing the cycle and is shown in Fig. 2. G and F mean generator, and on the other hand, D_X and D_Y mean discriminator. The basic learning method is the same as GAN, but the added point is that the training is further progressed so that F(G(X)) equals X. The opposite direction is also the same. In addition, the discriminator is also learned by using $D_X(X)$ and $D_Y(G(X))$ to determine whether it is a generated image or not.

The biggest feature of CycleGAN lies in the cyclic structure and dataset. The title of the paper, in which CycleGAN is introduced, is "Unpaired image-to-image translation using cycle-consistent adversarial networks". As the title indicates, an unpaired dataset is used. Most networks are trained using paired datasets. However, in real life, there are less paired datasets than unpaired datasets, so CycleGAN can solve this problem [6]. Therefore, collecting a dataset itself is easier than before. CycleGAN is used for style transfer, for example, summer-winter transfer, photo generation from paintings, and photo enhancement, etc.

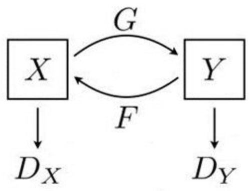

Fig. 2. CycleGAN structure [6].

B. *Segmentation*

Segmentation is one of the key fields in computer vision, and it is difficult task that requires not only classifying images, but also understanding images. Segmentation is divided into semantic segmentation and instance segmentation. Semantic segmentation determines which class each pixel belongs to and does not distinguish subordinate objects of the same class. Representative models of semantic segmentation include FCN [2], BiSeNet [3], and DeepLab [4].

Instance Segmentation is a method of determining as a different instance even if it is in the same class. It requires a larger amount of computation than Semantic Segmentation. Representative models of instance segmentation include Mask R-CNN [7] and YOLACT [8]. Fig. 3 shows the difference between semantic and instance segmentation.

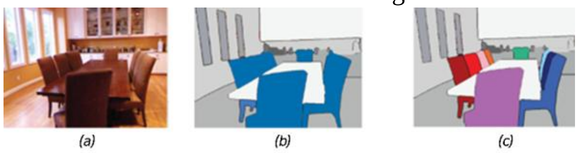

Fig. 3. Results of Segmentation Methods
(a) Input image, (b) Semantic segmentation image,
(c) Instance segmentation image

BiSeNet [3] has a bilateral structure and is shown in Fig. 4. It is a structure consisting of two paths: Spatial Path (SP) and Context Path (CP). CP uses a pre-trained Xception model. This model enables real-time semantic segmentation by combining SP and CP. Accuracy and processing speed varies according to the pre-trained model used for CP. In the case of BiSeNet using the Xception 39 model, processing speed was 105.8 (fps) and mIoU were 68.4(%) in the Cityscapes test dataset. When the Res 18 model was used, processing speed was 65.5 (fps) and mIoU were 74.7(%) [3]. It shows that there is a trade-off between accuracy and speed.

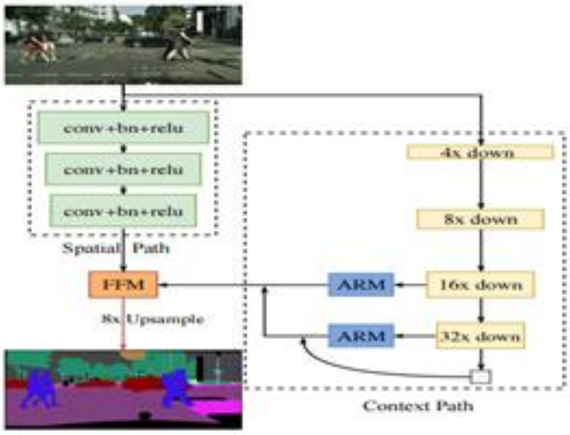

Fig. 4. BiSeNet Structure [3].

In CP block, sufficient receptive field should be provided. However, in SP block, the combination of convolution, normalization, and activation function is repeated, and the spatial size of the original input image is preserved, and spatial information is encoded. Since CP and SP are calculated at the same time, the efficiency is increased, and they are used complementarily for high performance. In SP, it was used to obtain spatial information by reducing the size of the input image. It was used to find an object using a feature extractor learned by a pre-trained model in CP. The results of SP and CP are merged in the Feature Fusion Module (FFM). The results of the SP have a wealth of detailed information, and the results of the CP have contextual information. This means that SP has a low-level feature and CP has a high-level feature. Since the results of SP and CP have features of different levels, it cannot be used immediately. Among the style transfer methods described above, Segmentation was added to the Deep Photo Style Transfer method [9] and the system is similar to our method. Deep Photo Style Transfer is a method that came out to solve the problem that the Style Transfer method could not maintain the shape of the original image, but we want to get photo realism. Segmentation information was added using Pyramid Scene Parsing Network (PSPNet), so that only the area corresponding to each class was changed in style. In segmentation, the part that looks like noise or similar class uses the grouping method to reduce the class, making more detailed segmentation [9]. However, it had a complex model and structure for the above process, which required a lot of processing time. According to the article [9], when using NVIDIA GeForce GTX 1080, it took about 16 minutes to generate one image. Therefore, it is difficult to operate in real time and there needs improvement. Also, results are not good for complex scenes. Our method showed better results compared to the existing method [10].

C. *Group Normalization [10]*

One way to solve the problem that occurs when the batch size is small is Group Normalization (GN). GN is a method of normalizing each channel by dividing it into N groups. Like the existing image processing techniques such as histogram of gradient (HOG) method and scale invariant feature transform (SIFT) algorithm, the focus was on the method of dividing the features of the image into several groups [10]. Several normalization methods are shown in Fig. 5.

And the result of comparing with ImageNet Validation errors for each normalization method is shown in Fig. 6 [10]. Also, in the case of GN, the performance varies depending on the number of groups which is set as hyper-parameters. Since it may vary depending on the data set or network model, it must be set through experiment.

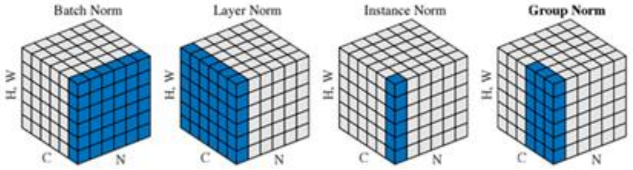

Fig. 5. Normalization Methods.

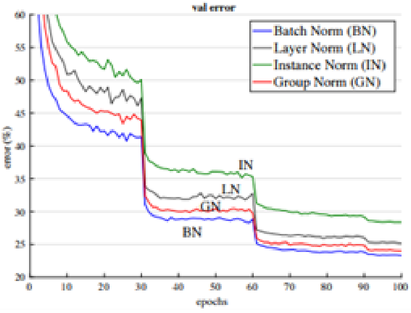

Fig. 6. ImageNet Validation Error Comparison of Normalization Methods [10].

III. PROPOSED STYLE TRANSFER MODEL

A deep learning model with the purpose of separating a specific area in moving video using BiSeNet and transferring the area to a specific style using CycleGAN in real time is proposed as shown in Fig. 7. Since the structure of the proposed model has a parallel structure, it is possible to keep the efficiency of computing resources if each model is operated in two GPUs to process each operation. Even though this is possible with one GPU, however, because two models must utilize the result at the same time, running the computation by dividing one resource can cause a slowdown.

BiSeNet was used as the segmentation network. The advantage of BiSeNet is that because it uses a pre-trained model, it is possible to expect a certain degree of accuracy and that it is possible in real time with a simple structure [11].

In this paper, at CycleGAN and BiSeNet block, group normalization (GN) was used instead of batch normalization (BN). Training was performed with different coefficient loss, which is a parameter how much context of the original will be left. The larger the coefficient loss, the more the original is maintained. Because subjective judgment is involved, you should find the optimal value for the best subjective quality.

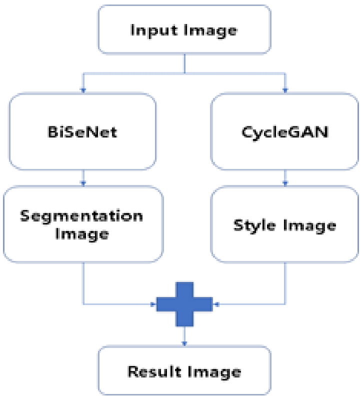

Fig. 7. Proposed specific area style transfer model.

To train the CycleGAN, you need to put the necessary data in generator and discriminator network. Since the discriminator is needed during training time only, the waste of resources due to the discriminator should be reduced during the test. Also, since the generator has a circular structure, it should be set in one direction rather than in both directions. Training images are resized to (256, 256). In the case of BiSeNet, grouping was used to use only the necessary part of the dataset. roads, sidewalks, and lanes were grouped into one target class and the rest were treated as background. In this case, all classes related to road could be detected. GN applied only to the ARM part in BiSeNet. ARM is a part that combines the result of the pre-trained model in CP, and the purpose of CP is to detect an object. It was assumed that the result would be good if the part of detecting the object and combining the result was done more accurately. The reason why GN is not used for SP is that it is determined that the training accuracy of CP is higher than that of SP.

IV. TRAINING PROCEDURE

A. *Experimental setup*

The experiment setup is as follows. Desktop PC has 16G RAM and is equipped with two RTX 2080 T_i GPU boards. To operate CycleGAN and BiSeNet together in real-time, each module is running separately on each board. The OS was Ubuntu 16.04, and the used libraries were Python 3.5, CUDA 10.1 and TensorFlow 1.15.

B. *Training Dataset*

To proceed style transfer and segmentation together, it takes a long time because the network structure tends to become deep and complex. Therefore, each model must have a lightweight structure and obtain satisfactory performance with accuracy. Training for style transfer proceeds with unpaired data. Dataset consists of Grass and Road images. CycleGAN plays the role of changing the Style to Grass for segmented Road area. The dataset required for the training of CycleGAN was collected directly using internet crawling. However, since many of the data are composed of irrelevant or difficult data to use, we should choose appropriate ones. It consists of Road and Grass images, and each training data consists of 439 and 243, respectively, and test data consists of 100 each.

In the case of road image, it was made up of roadways, sidewalks, and car lanes, so we collected road images more than grass images. Samples of the collected dataset is shown in Fig. 8.

Fig. 8. Unpaired Grass & Road Data Set.

To train BiSeNet, video CamVid dataset [12] is used. Since most of the classes are limited to what can be seen on the road, we chose this dataset composed of 33 classes. An example of CamVid dataset is shown in Fig. 9.

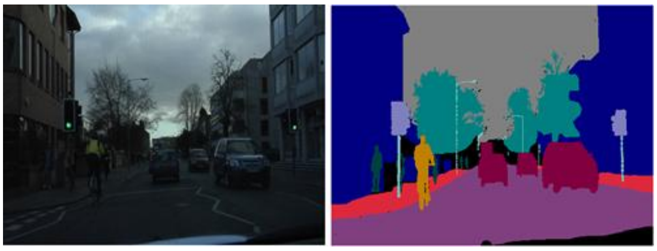

Fig. 9. CamVid Data Set (Original Image (Left), Ground Truth (Right)).

V. RESULTS AND DISCUSSION

A. *Style Transfer Results*

Since the result value of CycleGAN is expressed as a value between [-1, 1], post-processing was used to change the value between [0, 255] by using the mapping method to obtain the value of the original image. In the case of the mapping method, following equations were used.

$$\mathrm{Min_o} = -1,\ \mathrm{Max_o} = 1$$

$$\mathrm{Min_n} = 0,\ \mathrm{Max_n} = 255$$

$$\mathrm{Value_o} = \mathrm{input}$$

$$\mathrm{Range_o} = (\mathrm{Max_o} - \mathrm{Min_o}),\ \mathrm{Range_n} = (\mathrm{Max_n} - \mathrm{Min_n})$$

$$\mathrm{Value_n} = (((\mathrm{Value_o} - \mathrm{Min_o}) * \mathrm{Range_n}) / \mathrm{Range_o}) + \mathrm{Min_n}$$

For example, if input value is 0.7, the final value is 216. This value is used by converting the type to Integer. The training was conducted by varying the coefficient loss from 0 to 100. The result of the network corresponding to each loss is shown in Fig. 10. As the coefficient loss increases, the original shape is maintained. Fig. 10 shows the original road image, grass-styled road image, and reconstructed road image from styled road image from left to right at each coefficient loss.

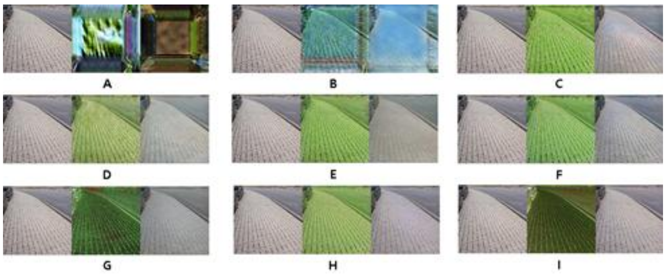

Fig. 10. Results with various coefficient loss (A:0, B:1, C:5, D:7, E:10, F:20, G:30, H:50, I:100).

In Fig. 10, each training took about 2 hours and 40 minutes. When the coefficient loss was 30 in Fig. 10(G), we thought that it was the most grass-like by subjective judgment.

Experiment was performed with the coefficient loss being 30. Fig. 11 shows the result when coefficient loss is set to 30. Each image shows an original image, a style transferred image, and an image in which the style transfer image is re-transferred to original road style. The larger the loss value, the smaller the difference between the original image and the re-transferred image.

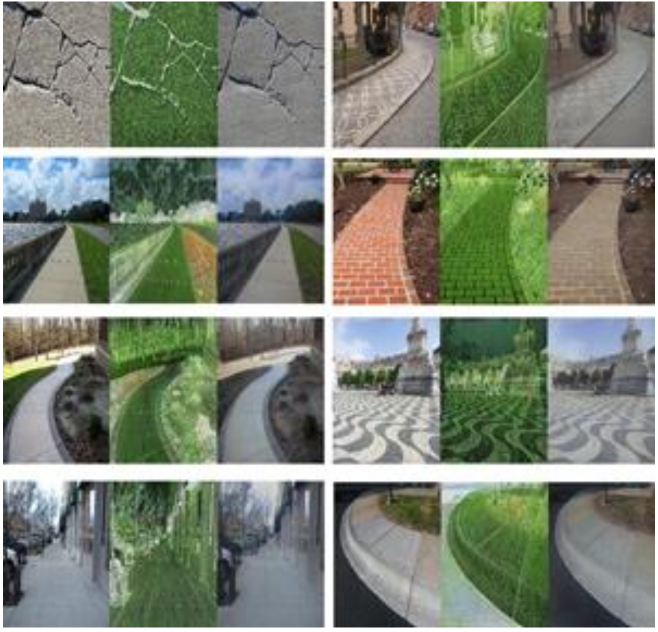

Fig. 11. CycleGAN Result (Coeff. Loss = 30).

B. *Segmentation Results*

In CamVid Data Set consisting of 33 classes, 5 classes to be judged as Road class were selected to have a label of (255, 255, 255). A background with a label of (0, 0, 0) has a black value, and an object with a label of (255, 255, 255) has a white value to use when accessing pixels. The data set consists of about 700 frames of road images. BiSeNet results should be divided into two types of road and background. Therefore, data is expressed in binary as described above. As you can see in Fig. 12, the road is represented in white, and the background is black, and the contour of the road is taken and overwritten in the original image.

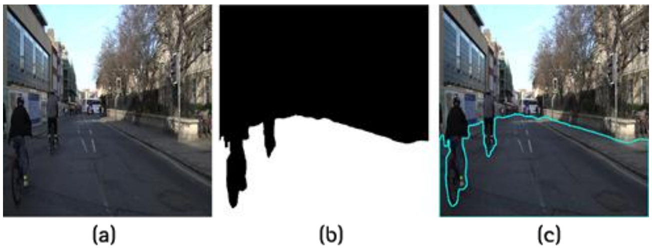

Fig. 12. BiSeNet Result. (a) Original Image, (b) Segmentation result (Road (White), Background (Black)), (c) Combined result

In this paper, as CP's pre-trained model in BiseNet segmentation method, Xception model was replaced by ResNet-101 model, which was used for the purpose of improving the accuracy of segmentation. As a result of the experiment, mIoU was 71.4(%) when Xception model was used to Cityscapes dataset, and mIoU was 78.9(%) in case of ResNet-101. In the COCO-Stuff validation Data Set, when the Xception model was used, mIoU was 22.8(%). However, replacing ResNet-101 model, we obtained mIoU value of 31.3(%) [3]. ResNet-101 model showed about 8(%) performance improvement. In terms of speed, it shows that the processing speed is 105.8 (fps) when using the Xception model, while 65.5 (fps) when using ResNet-18. In case of replacing ResNet-101, the result was about 12 (fps) through an experiment.

Training method is changed from the Naive model using BN of BiSeNet to the modified model with different number of GN groups. The number of groups was 4, 8, 16, and 32, and training was conducted. When training with the number of groups of 16 and 32, training was conducted within half of the epoch. The reason is that as training progresses, the convergence of accuracy has increased.

From the Naive method using BN, the FFM part in BiSeNet was trained with CamVid dataset by varying the number of GN groups. The training results of BiSeNet are shown in Table 1. Proposed GN methods took less training time than the Naive method, and you can see that the Precision and IoU have improved.

Looking at the table, in training phase, you can see that IoU is the highest when G is 8. In case that G is 16 and G is 32, we trained 150 (epoch), so if we trained up to 300 (epoch), it seems that it will increase numerically.

Table 1. BiSeNet Training Results.

Method	Precision (%)	IoU	Training time
Naïve BN	90.75	52.89	15h (300epoch)
GN_4	91.18	53.21	11h (300epoch)
GN_8	91.21	53.67	11h (300epoch)
GN_16	90.99	52.03	5h (150epoch)
GN_32	91.32	51.21	5h (150epoch)

In test phase, when G is 8, the test accuracy is 97.73(%) and the IoU is the highest at 73.34. The execution time is 0.035 (sec/frame), showing that real-time operation is possible. Segmentation results are shown in Fig. 13.

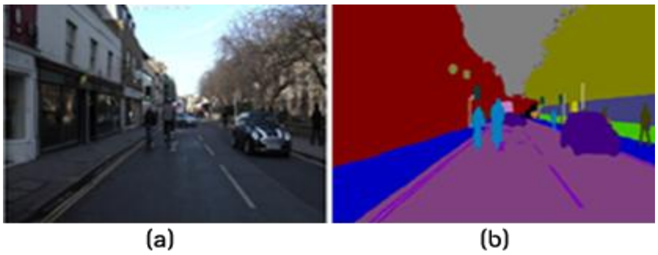

Fig. 13. Segmentation result of CamVid Data ((a) Original frame, (b) Labeled frame).

To make the segmentation more accurate, the erosion and contour restrict method were applied. Erosion is a method that converts to make the image to the lowest value inside the filter by eroding it by using the determined kernel size.

This makes it possible to create a blurred border as a background. If you only use the erosion method, you will get an eroded image. However, it was complemented by using the contour restrict method together. If you use the contour restrict method to sort the contours according to their size and import only the large ones, the contours that are considered noise will disappear.

C. *Grass style transfer to segmented Road in real-time video*

We need to know the coordinates in segmentation to create the final output by collecting the results obtained through CycleGAN and BiSeNet. In case of using the Pixel Access method, it takes a long time because the pixel value corresponding to the style image must be obtained after accessing each pixel and determining whether it is a background or an object. On the other hand, the Split Access method is fast because it uses color to separate the background from the object. After performing segmentation for RGB three channels, only one channel value is compared for speedup. As shown in Table 2, the processing time is reduced by about 4 times.

Table 2. Processing Time Comparison by Segmentation Judging Method.

Method	Time(sec)
Pixel Access	2.066
Split Access	0.554

To get as close to real-time as possible, one GPU was separately allocated to CycleGAN and BiSeNet and executed simultaneously. Fig. 14 shows the experimental results. You can see that the Road has changed into Grass Style.

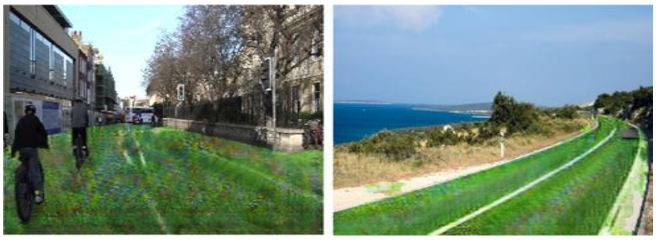

Fig. 14. Results of the proposed model.

Retrieval Number: 100.1/ijitee.E86890310521
DOI: 10.35940/ijitee.E8689.0310521
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

When testing with a real-time video, the processing speed appears to be about 6(fps). Although this is lower than 30(fps), if the processing time is reduced by pixel level operation, it will be closer to real-time.

Fig. 15 is video test results which were applied to various types of video. Experiments were conducted on three different kinds of video. In Fig. 15(a), inside the video with a lot of people, it can be confirmed that it is generally good except for the area covered by people and the area of the stairs.

In Fig. 15(b), in case of the roadway, if vehicles, people, and bicycles are detected and style transfer is applied well only to the road. In Fig. 15(c), in case of sidewalks, only areas that were thought to be roadways were detected, but the sidewalk area was not detected well.

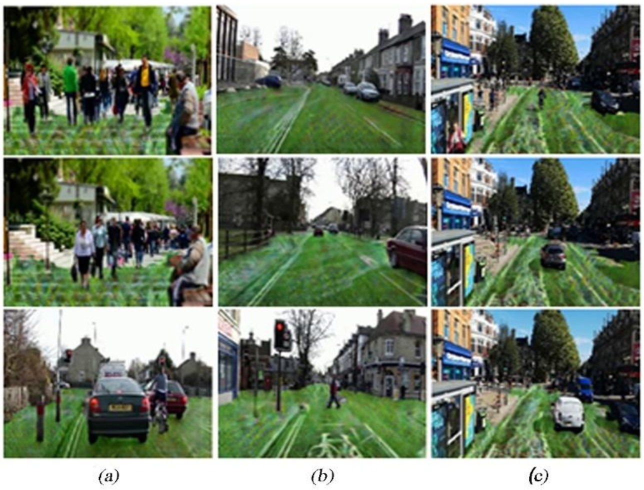

Fig. 15. Video Test Results. (a) lots of people in the road, (b) roadways, (c) sidewalks with bus stop.

Fig. 16 shows the failure examples. You can see the segmentation is not correctly done. The probable reason is that there was lack of training data. If more training data are provided, the style transfer performance will be improved quite a lot.

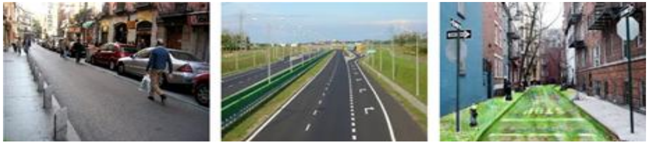

Fig. 16. Failure examples.

VI. CONCLUSION

In this paper, we propose a style transfer on a specific area in real time video. We improved performance and speed using Group Normalization in the segmentation model that finds a specific area and the model that performs style transfer. For a trade-off between performance and speed, we changed the pre-trained model of the BiSeNet model to the Res101 model, with increasing the performance and slightly lowering the speed. To use CycleGAN and BiSeNet at the fastest speed, it was made to operate on two GPUs, and because of testing with a video, the processing speed was measured as 6 (fps). The overall speed of obtaining the result has decreased due to BiSeNet. The CycleGAN used for style transfer was trained using the directly collected dataset by crawling, and the style was transferred to apply the style of Grass to the Road image. Training speed was improved by using color information, and the segmentation accuracy was also improved by using GN instead of BN. By changing the number of groups in GN, training was conducted to find the optimal number of groups, and it was confirmed that the performance was not decreased significantly even if the epoch was reduced in half. In the case of BiSeNet, since Res-101 model was used instead of the Xception model for the pre-trained model, the accuracy was increased but the speed was decreased. Erosion and contour restrict method were used to improve against inaccurate segmentation. The CamVid dataset is used for training, and performance is good in most road images.

In the case of CycleGAN, the number of datasets to be trained was about 900, which was not sufficient, so it did not show the best image generation results. Performance can be improved by increasing the number of training data. In the case of BiSeNet, the performance was not good in the case of images with many hidden parts. Since it was not possible in real time due to the slowdown due to the post-processing of BiSeNet, it is necessary to improve the pixel access method for post processing.

ACKNOWLEDGMENT

This research was supported by the Ministry of Land, Infrastructure, and Transport (MOLOT, KOREA). Project ID: 21TLRP-B146733-04, Project Name: Connected and Automated Public Transport Innovation (National R&D Project).

REFERENCES

1. L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” *arXiv:1508. 06576v2*, September 2015.
2. J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *arXiv:1411.4038*, 2015
3. C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “BiSeNet: bilateral segmentation network for real-time semantic segmentation,” *arXiv:1808.00897*, August 2018
4. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *arXiv:1606.00915*, May 2017
5. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *arXiv:1406.2661*, June 2014
6. J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *arXiv:1703. 10593*, March 2017
7. K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *arXiv:1703.06870*, March 2017
8. D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “YOLACT: real-time instance segmentation,” *arXiv:1904.02689*, Dec. 2019.
9. S. Penhouët and P. Sanzenbacher, “Automated deep photo style transfer,” *arXiv:1901.03915*, Jan. 2019.
10. Y. Wu and K. He, “Group normalization,” *arXiv:1803.08494*, Mar 2018
11. G.T. Kim, Y. Lee, H.H. Ko, and K.H. Lee, “A study on the improvement performance of objective segmen- tation using group normalization method,” *J. of adv Research in Dynamical & Control Systems*, vol. 11, pp.2439-2445, Special Issues-05, 2019.
12. Motion-based segmentation and recognition(camvid) dataset, Available: http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/ 2008.

AUTHORS PROFILE

GeunTae Kim, received Bachelor of Engineering degree and master's degree from Dept. of Electronics and Communication Engineering at Kwangwoon University in 2018 and 2020, respectively. He is currently researcher in ABH inc. His research interest is in Image Processing, Object Detection, Face Emotion Recognition and Style Transfer using Deep Learning.

Hyunmin Kim, received a Bachelor of Science in Computer Science in 1992 and a master's degree from Dept. of Electronics and Communication Engineering at Kwangwoon University in 1996. He is currently the Research Director at TELECONS, Inc. His research interests are autonomous driving, SLAM, object detection and image processing.

Hyung-Hwa Ko, is a Professor at Kwangwoon University. He received Bachelor of Engineering Degree, Master's Degree and Ph. D from Seoul National University. He joined Kwangwoon Univ. as a faculty member from 1985. His research interest is Information theory, JBIG2, H.264/AVC, HEVC, Machine Learning, Deep Learning applications, and Autonomous driving.