Semantic Segmentation and Synthesis for Paintings

Zheyu Liu, Zheying Lu, Qinghui (Luna) Xia

Abstract

What were the views that artists saw 500 years ago? Curious about the sceneries depicted in artworks, we propose a deep learning approach to reconstruct photo-realistic views from paintings. Models that we used in our project were DeepLab (Chen et al., 2016) for semantic segmentation on photos and SPADE (Park et al., 2019) for photo-realistic image synthesis. The novelty of our project focused on improving segmentation performance on landscape paintings. Given that most of the current segmentation techniques and datasets were based on real photos, we conducted data pre-processing by applying style transfers to landscape photos with the AdaIN model (Huang and Belongie, 2017). Our approach achieved reasonably accurate reconstructions for some landscape paintings, as shown in both quantitative and qualitative evaluations.

1 Introduction

With developments in Artificial Intelligence and the rise of interdisciplinary practices in the art industry, it was without doubt that AI-infused artworks had come into view. Machines gained the ability to create music, paintings, photos, and even a novel that won a literature prize (Smithsonian Magazine, 2016). As a new addition to traditional art-making tools, AI-Art also spawned controversy. Danny (Smithsonian Magazine, 2016) stated that such technologies could bring inspirations to authors. But in the long term, replying on machines for inspirations might negatively impact human creativity. Another aspect of the conversation raised doubts over the authenticity of machine-generated articles. How to assess similarities between the generated results and the training sets was a challenge with great moral values.

Sharing strong interests in machine-generated artworks, we researched the applications of Generative Adversarial Networks (GAN) (Creswell et al., 2018) in such fields, and attempted to find a non-controversial research direction.

There were a lot of existing researches on transforming photos using different styles. Inspired by A Neural Algorithm of Artistic Style (Gatys et al., 2015), which performed style-transfer on a realistic photo to get an artistic photo, we obtained the idea of converting paintings to realistic photos to get a sense of what the painted views would look like in real life. Besides the generation part, evaluation was also an essential task for us. CAN (Elgammal et al., 2017), a model that combined different artistic styles in generating art pieces, proposed quantitative and qualitative metrics to determine if a piece of artwork met certain criteria. Part of their metrics were realized by conducting surveys with groups of people, which was a method we adopted in qualitative assessments.

2 Related Works

CNN Convolutional Neural Networks (Lecun et al., 1998) had been widely applied to computer vision tasks like image classification and image segmentation. Albawi (Albawi et al., 2017) stated that peoples' interests in deeper hidden layers had recently begun to make a great impact on the performance of classical methods in different fields, especially in pattern recognition. With an increasing amount of researches conducted on deep models, GANs (Creswell et al., 2018) gained attention with their excellent performance in computer vision tasks.

Generative Adversarial Models Generative adversarial networks (GANs) (Creswell et al., 2018) are capable of learning deep representations without extensively annotated training data. GAN applications include image synthesis, semantic image editing, style transfer, and image classification. A lot of image generation related topics have conducted researches with GANs, such as the

famous fake face generator (Karras et al., 2018). We also used a GAN for data pre-processing. GAN-related works inspired us to look deeper into image segmentation and image synthesis tasks.

Image Segmentation Image segmentation is a key task in computer vision (Minaee et al., 2021). The broad success of Deep Learning (DL) has prompted the development of new image segmentation approaches leveraging deep learning models. A lot more different architectures arose in this field of study. With the existing architectures, we were able to conduct reasonably accurate image segmentation for real photos. However, we did not find any existing work on segmentation for paintings. In the following sections, we will illustrate more on our approach to solve this problem.

Image Synthesis Developments in Image Synthesis tasks have brought research understandings of image contexts and styles to another level. SPADE (Park et al., 2019) proposed an approach for generating realistic and high-quality photos given simple hand-drawn lines as inputs. We included the model in our project to explore its broader potential in applications.

3 Datasets

For the training dataset, we chose COCO-Stuff (Caesar et al., 2016) as our base dataset and generated Landscape-Dataset and AdaIN-Dataset from COCO-Stuff.

3.1 Training dataset

Our primary goal of this project was to generate synthesized photos from paintings. We planned to train a segmentation model for paintings, and feed its results to SPADE (to be discussed in the following section). Given that SPADE generated synthesis outputs based on pixel-wise semantic segmentation masks from the segmentation model, the segmentation labels (for example, 1 for sky, 2 for trees, etc.) between the two models should match. In the following sections, we will discuss the details of our data-prepossessing part.

3.1.1 COCO-Stuff

In this project, we selected COCO dataset (Lin et al., 2014) as the dataset for both models to share consistent labels. After observing inference results from DeepLab using different types of paintings, we found that this segmentation architecture performed better on landscape objects. Also, a lot of paintings were about views or landscapes. The COCO-Stuff dataset (Caesar et al., 2016) augmented all 164K images from the popular COCO dataset with pixel-level stuff annotations. These annotations can be used for scene understanding tasks like semantic segmentation, object detection and image captioning.

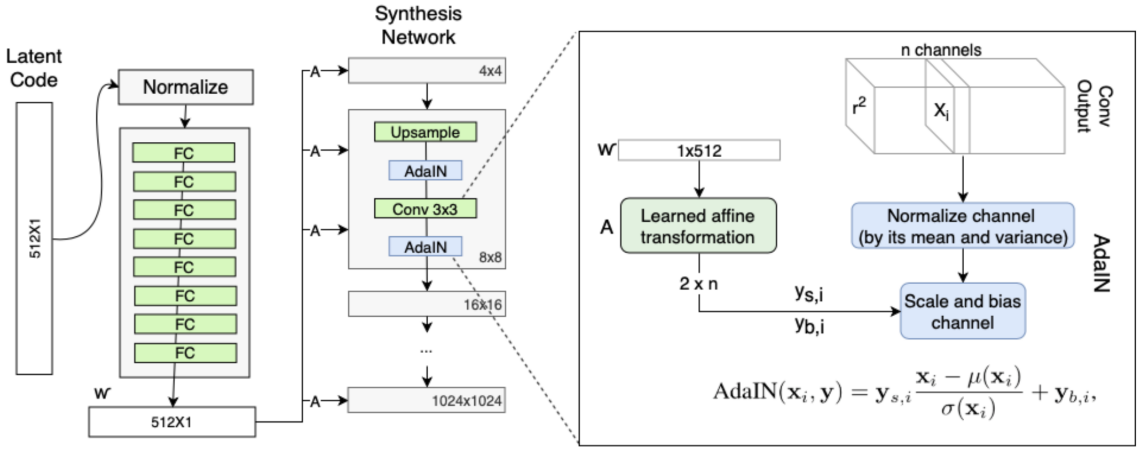
Figure 1: Landscape-Dataset

3.1.2 Landscape-Dataset

Considering of the observations mentioned before, we decided to select landscape photos from the original dataset to construct our Landscape-Dataset. Across all 171 categories in the COCO-Stuff dataset, we manually selected 33 categories as common features of landscape photos. If images had more than five landscape features, we included them as landscape images. By doing this step, We were hoping to better familiarize or bias the segmentation model with landscapes. This dataset consisted of 8191 training samples and 900 testing samples.

3.1.3 AdaIN-Dataset

After compiling Landscape-Dataset, which was a subset extracted from COCO-Stuff dataset with landscape photos, we focused more on the differences between paintings and photos. The existing architectures for segmentation tasks all experimented on real photos. However, we wanted to feed paintings into the model as our inputs. Since there was no existing dataset with pairings of paintings and annotation maps, we believed that conducting style-transfer on the COCO-Stuff dataset with its original annotations could be a solution to this problem. AdaIN (Huang and Belongie, 2017) is a tool for style transfer. It is similar to



The generator's Adaptive Instance Normalization (AdaIN)

Figure 2: AdaIN architectue

the architecture mentioned above - A Neural Algorithm of Artistic Style (Gatys et al., 2015). The AdaIN (Adaptive Instance Normalization) module transfers the encoded information, created by the Mapping Network, into the generated image. The module is added to each resolution level of the Synthesis Network and defines the visual expression of the features in that level:

1. Each channel of the convolution layer output is first normalized to make sure the scaling and shifting of step 3 have the expected effect.

2. The intermediate vector w is transformed using another fully-connected layer (marked as A) into a scale and bias for each channel.

3. The scale and bias vectors shift each channel of the convolution output, thereby defining the importance of each filter in the convolution. This tuning translates the information from w to a visual representation.

Here we chose AdaIN over other approaches is because it is faster and has more flexibility for users. The key to the efficiency in such feed-forward method is that AdaIN applies instance Normalization in each layer which normalizes the mean and standard deviation which are computed across spatial dimensions independently for each channel and each sample.

We applied AdaIN style-transfer on roughly half of our Landscape-Dataset to generate AdaIN-Dataset, which consisted of painting-styled images. The resulting dataset contained 3994 and 899 images in training and testing sets, respectively. By performing this step of data pre-processing, we were aiming to improve segmentation performance on paintings.

Figure 3: AdaIN-dataset

3.2 Test dataset for the pipeline

To better test our model, we divided the dataset into simple and complex categories depending on different painting styles. For the simple category, we included some of the realistic painting styles. And for the complex models, we included more abstract painting styles such as Impressionism. By dividing our test set into simple and complex categories, we were hoping to improve our model performance on a more specific branch, focusing on the simple baseline while reaching for achievements in the complex cases.

4 Methods

4.1 Overview

The base dataset we used was the COCO-Stuff dataset (Caesar et al., 2016). As mentioned above in the Datasets section, we extracted landscape photos from COCO-Stuff to form Landscape-Dataset, and later generated AdaIN-Dataset by applying style transfers to the landscape images.

The training process was conducted in 4 steps. First, we loaded a DeepLab model with pre-trained weights, and trained it on Landscape-Dataset. This step aimed to improve DeepLab segmentation performance on landscape photos specifically. We now had our DeepLab-Landscape model. Another variant of DeepLab, named DeepLab-AdaIN was developed similarly by training DeepLab on AdaIN-Dataset. This variant could help us assess whether a modified DeepLab could generate better segmentation results on paintings.

After training, we ran inferences on the DeepLab variants using the test set from AdaIN-Dataset as inputs. This would generate segmentation maps from synthesized paintings. We then fed the outputs into SPADE to get synthesized photos.

In the following sections, we are going to further discuss the models we used in this project.

4.2 Architecture

4.2.1 DeepLab

DeepLab (Chen et al., 2016) re-purposes a DCNN trained for image classification tasks to semantic segmentation tasks. The model overcomes three challenges.

First, DCNNs designed for image classification perform repeated pooling and downsampling at consecutive layers. This causes significantly reduced spatial resolution in feature maps. DeepLab computes the feature maps in a higher sampling rate by upsampling filters in convolutional layers.

The second challenge is to segment objects at multiple scales, for which DeepLab resamples a given feature layer at multiple rates prior to convolution.

The third challenge is caused by the built-in invariance of DCNNs, which is essential to local image transformations and object-centric classifiers. However, it reduces the spatial accuracy of dense prediction tasks. DeepLab manages to capture fine details by using a fully-connected Conditional Random Field. Figure 4 shows the pipeline of DeepLab for a semantic segmentation task.

4.2.2 SPADE

SPADE (Park et al., 2019) focuses on a specific form of conditional image synthesis which converts a semantic segmentation mask to a photo-realistic image; they refer to this as semantic image synthesis.

In conventional network architecture built for semantic synthesis tasks, a stack of convolutional, normalization and nonlinearity layers lead to sub-optimal results because the semantic information from input masks tend to get "washed away" by normalization layers. The solution is the SPatially-Adaptive (DE)normalization, or SPADE, which uses the input layout for modulating activations in normalization layers through a spatially-adaptive, learned transformation.

Specifically, the semantic segmentation mask is projected onto an embedding space, and convolved to produce modulation parameters. These are tensors with spatial dimensions rather than vectors. Then, the parameters are multiplied and added to the normalized activation element-wise. Each normalization layer uses the segmentation mask to modulate layer activations. Figure 5 shows the architecture of the SPADE generator, which uses a series of SPADE residual blocks.

5 Experiments

5.1 Naïve-DeepLab

Our first experiment started with the pre-trained DeepLab segmentation model and pre-trained SPADE model, which served as the baseline of our experiments for generating synthesized photos from the paintings. This naive pipeline showed that synthesis results were affected by both segmentation and synthesis models. The segmentation model defined the boundaries of each element in the input images, outputting labels and maps that would affect the faithfulness of SPADE generation results. More specifically, if the segmentation model performed poorly, such as classifying the sky as the mountain, even though the synthesis model generated mountains with high coherency and quality, the synthesized photo still would not match with the input paintings. To improve the SPADE model, as mentioned in (Park et al., 2019), a better model was trained on the Flickr dataset, which was composed of landscape photos, but both the dataset and the pre-trained model were not published.

By observing results from our baseline approach, we also found that synthesis quality for our specific

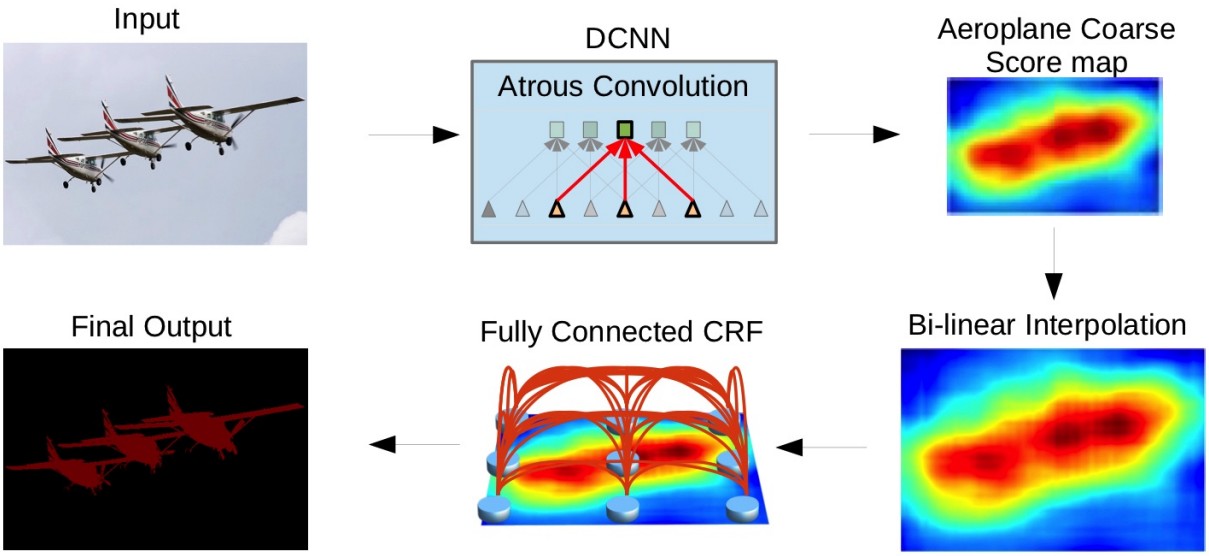


Figure 4: DeepLab architecture

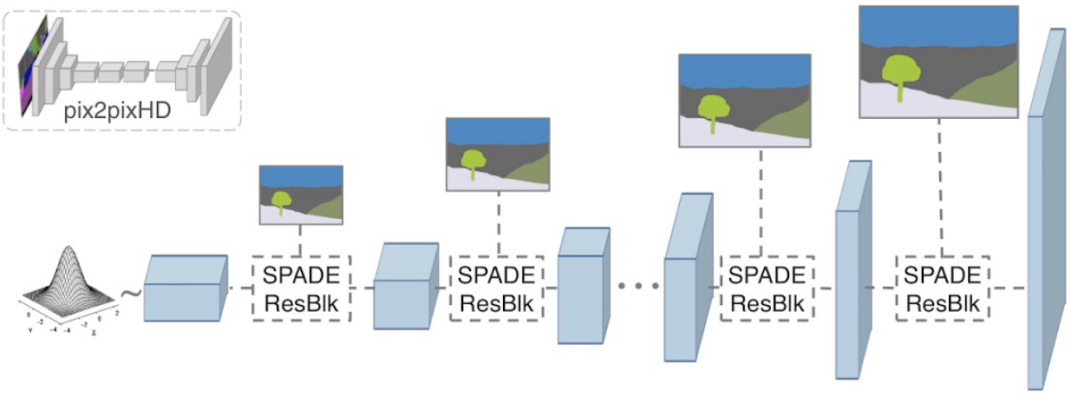


Figure 5: SPADE generator architecture

problem might be correlated with how detailed semantic segmentation was labeled in the datasets. For example, all types of trees, such as pines, willows, and maples, were labeled as trees, which resulted in a partial loss of information after the segmentation step. The synthesized photos hence contained inaccurate tree textures. Consequently, a dataset with more labels should help further improve the quality of the synthesized photos.

We first trained the segmentation model from scratch using Google Cloud Platform, and as mentioned in section 6.1, the experiments crashed after 800 epochs due to being out of CUDA memory. However, the IOU only increased to around 0.02, which is significantly lower than the released pretrained model on the COCO dataset with an IOU of around 0.32; we thus decided to use the released model from DeepLab, considering the hardware limitations and the time constraints.

As mentioned above, synthesis quality depended on both the segmentation and synthesis models. While the synthesis model could be improved through training on a Flickr dataset, the details were not released. Hence, our experiments focused on improving the segmentation model, and repurposing it to produce more accurate segmentation maps on AdaIN-Dataset.

5.2 DeepLab-Landscape

Our project aimed to synthesize photos from a set of paintings we selected, which mainly consisted of landscape contents. Since a large proportion of the COCO-Stuff dataset contained indoor objects, which was out of scope for our problem, we decided to experiment with a smaller and more specific dataset of landscapes. We envisioned a

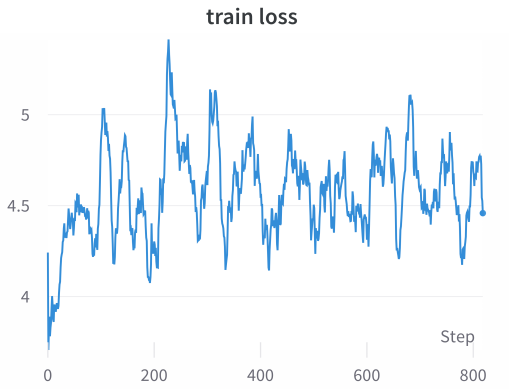


Figure 6: Deeplab-landscape training loss

Figure 7: Deeplab-AdaIn training loss

segmentation model more suitable to the problem, if we trained it on a dataset with themes that closely aligned with the common themes in our paintings. Therefore, we extracted a dataset with around 5000 images from the COCO-Stuff dataset following the approach mentioned in 3.1.2. As shown in the training loss plotted in figure 6, the training loss did not converge. We then further tested the saved model on the test set (composed of the landscape photos). The result in 6.1.1 indicates that training in these selected landscape photos did not improve the semantic segmentation on landscape photos.

5.3 DeepLab-AdaIN

The experiments described in 5.2 attempted to improve the model through training on a dataset whose styles and contents conformed with the paintings better. As mentioned in 3.1.3, we did style transfers using the AdaIN model on Landscape-Dataset extracted from COCO-Stuff. Our baseline pipeline, as mentioned in 6.1, worked well with paintings in the simple category, but yielded relatively poor results on the test dataset. We thus decided to investigate if training on a dataset of Impressionism-like paintings would help the model better identify the objects in paintings. Given that the annotations of the painting segmentation were rather expensive, we used AdaIN to automate such a style transfer process on the landscape dataset because the annotations of the segmentation were already available. The training loss of this experiment shown in figure 7 indicates that the training loss converged from 10 to 6 after 1000 epochs.

6 Evaluation

6.1 Single Images

6.1.1 Quantitative Evaluation

Following the pipeline described above, we trained several models and compared their testing performances with metrics provided by DeepLab (Chen et al., 2016). The percentage measures of the mean pixel intersection-over-union (mIOU), frequency weighted IOU, mean accuracy, and pixel accuracy across all dataset classes on Landscape-Dataset are shown in Table 1. The same metrics from testing on AdaIN-Dataset are shown in Table 2.

We evaluated the naive DeepLab model with and without its pre-trained weights as baselines for model comparisons. A direct evaluation of the pre-trained DeepLab yielded 35.99% and 14.30% on our landscape and AdaIN-processed datasets respectively. This set the highest performance on vanilla landscape segmentation, and second highest on non-photo images (paintings) following our guess that DeepLab performance would degrade in such cases. Training DeepLab from scratch using the landscape dataset seemed to quickly saturated to a high loss, resulting in a significantly reduced testing mIOU value of 2.505%. After noticing the performance drop, we decided only to used DeepLab with pretrained weights in developing variants.

Table 1 and 2 also show the testing performance of our landscape- and painting-tuned DeepLab variants. Our original intention behind fine-tuning DeepLab by re-using a subset of COCO-Stuff in the training set was to increase its exposure to landscapes. However, this variant did not outperform the baseline, possibly because the landscape dataset was too small comparing to what DeepLab was

	mean-IOU	frequency weighted IOU	accuracy	pixel accuracy
Naïve-DeepLab-Pretrained	**35.99**	**59.64**	**44.72**	**71.69**
DeepLab-Landscape	32.97	53.33	41.34	69.04

Table 1: Mean IOU, frequency weighted IOU, mean accuracy, pixel accuracy (%) of different models tested on Landscape-Dataset. The original DeepLab model with pretrained weights (Naïve-DeepLab-Pretrained) achieved the highest mIOU.

	mean-IOU	frequency weighted IOU	accuracy	pixel accuracy
Naïve-DeepLab-Pretrained	14.30	27.99	21.02	38.14
DeepLab-Landscape	13.16	25.39	19.41	37.86
DeepLab-AdaIN	**23.46**	**46.75**	**30.83**	**63.19**

Table 2: Mean IOU, frequency weighted IOU, mean accuracy, pixel accuracy (%) of different models on AdaIN-dataset. Our re-trained DeepLab variant (DeepLab-AdaIN) performed the best.

(a) (b)

Figure 8: Naïve-DeepLab pipeline on simple paintings

trained on. DeepLab-Landscape performed worse on paintings due to its compromised segmentation accuracy. We therefore changed directions and designed the DeepLab-AdaIN model variant independently from the Landscape dataset. This variant directly trained DeepLab on the style-transferred landscape paintings generated using AdaIN, and actually achieved a 1.64× improvement on mIOU when compared to vanilla DeepLab on paintings.

6.1.2 Human Evaluation

Our Naïve-DeepLab approach can achieve our goal in simple paintings where the objects are close to the real-world objects as shown in figure 8. However, it does not work well for more paintings, such as impressionism paintings, whose object textures or colors differ from the real world. To evaluate if the deepLab-AdaIN model improves the result with the complex category of the paintings, human assisted evaluation is introduced.

The quantitative evaluation where we compared the metrics between the experiments mentioned above claims a significant improvement after we trained the segmentation model on the style transferred landscape photos dataset, which indicates its potential in better synthesizing photos from the paintings. However, to confirm the improvement of the segmentation model trained based on the style transferred landscape photos dataset, we leveraged human assistance for the evaluation. Figure 9 shows a few randomly selected examples from our test result, wherein each example, photo 1 is the result denoted by the naive deepLab pipeline as mentioned in 5.1, and photo 2 is the result denoted by the deepLab-AdaIN pipeline as mentioned in 5.3. We then designed five questions using Google form to collect responses on the question, to what extent these synthesized photos resemble the original painting. We asked the users to select from the five options; photo 1 resembles better significantly, photo 1 resembles better slightly, cannot tell, photo 2 resembles better significantly, and photo 2 resembles better slightly. To analyze these results quantitatively, we encoded these options as -1, -0.3, 0, 0.3, 1, where the vote for significantly better is considered as one vote for the corresponding pipeline, and the vote for slightly better is considered as a partial vote. In consequence, if the average of the responses on an image is positive, the majority of the responses believe that the photos synthesized by the deepLab-AdaIN pipeline as mentioned in 5.3 are better; otherwise, the photos synthesized by the naive deepLab pipeline are better. The average score for each example is shown in figure 9 is shown in the following Table 3. We can then find that all examples have a positive average vote score, which indicates that most of the responses believe that the synthesized photos from these paintings using the approach defined in 5.3 is better than the approach defined in 5.1 for all of these examples. The second row and the third row in Table 3 indicate the percentage of responses in favor of the synthesized photos from the approach defined in

Figure 9: Comparisons of synthesized photos from naïve deepLab pipeline and deepLab-AdaIN pipeline

	example a	example b	example c	example d	example e
avg vote score	0.02	0.505	0.1	0.625	0.34
% responses in favor of photo 2	35	85	50	90	70
% responses not in favor of photo 1	50	90	60	90	85

Table 3: Human assisted evaluation

5.3 and the percentage of responses not in favor of the synthesized photos from the approach defined in 5.1. According to these examples, the most controversial one is the example (a), as shown in the figure 9, and we discussed with some of the interviewees who believe the photos generated using the naive approach are better; these interviewees believe that the trees in the distance are not correctly identified in the photos synthesized using the deepLab-AdaIN pipeline. However, from us as developer's viewpoint, the style transfer has resolved the problem in the naive pipeline, where the sky is misclassified as the mountain due to the texture of the painting. Although the trees in the distance are classified as buildings, these unusual shapes of the trees were not seen in the training set. Thus we still consider the example (a) as an improvement.

6.2 Images in Sequence (Video Frames)

In addition to the synthesized photos from single paintings, we explored photos synthesis from a sequence of frames in animated videos as another use case. Each frame in the animated videos could be classified as an simple category instance in our test dataset. This experiment is to explore if the synthesized photos from the video clips are consistent with their shared area.

As shown in figure 10, although the segmentation model, even using the naive approach, can segment the animated video frames with relatively high quality, this experiment still indicates two potential problems of approach. The first problem is that some of the information is lost after the segmentation; for example, ravines between the mountains are not generated in our result because this information is lost after the segmentation stage. Such a problem might be solved if the labeling on the segmentation is more abundant, given that there is no label like ravines on the COCO dataset. Additionally, when we generate a new video based on a sequence of synthesized photos, it is not as consistent as the original animated video. To be more specific, only the top part of the mountain changes across different frames. The rest of the mountain is almost not changing, which is reasonable because our training was not trained on any sequences of the video clips, and the consistency between the video frames was not considered in our architecture.

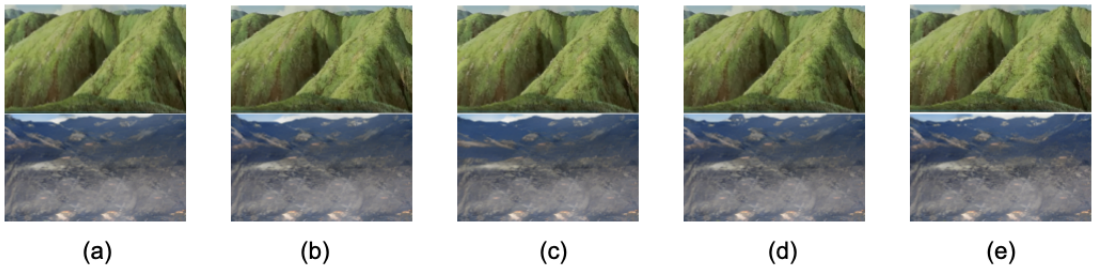


Figure 10: Synthesized photos from a sequence of animated video

7 Future Works

For future works, we would like to continue to improve the quality of the photos synthesis from paintings by resolving the current issues in our pipeline. For example, we may add another tone transfer stage after the SPADE outputs so that the tone of the synthesized photo would be closer to the original paintings. In addition to that, the experiments mentioned in 6.2 exposed a problem in a sequence of video frames, and we would like to resolve this issue by training with a sequence of images for the SPADE model.

8 Conclusion

To sum up, our initial approach with the naive segmentation model and the SPADE model worked for paintings depicting objects that were close to real-world objects. However, since more paintings lacked such nice traits, our naive approach performed relatively poorly on them. We thereby trained our segmentation model using style-transferred paintings, and improved the quality of the synthesized photos. Given a shortage of existing datasets with segmentation annotations for paintings, as well as the high costs of manual segmentation, our segmentation model did not use human-created paintings with segmentation for training. With a dataset consisted of machine-generated style-transferred photos, our model managed to synthesize photos from paintings with relatively high quality.

9 Appendix

[1] Part of our approach is derived from the following implementations:

https://github.com/NVlabs/SPADE; https://github.com/kazuto1011/deeplab-pytorch; https://github.com/naoto0804/pytorch-AdaIN

[2] Please refer to our wandb project for the loss curve and the evaluation metrics:

https://wandb.ai/qx2217/painting-synthesis/table?workspace=user-qx2217

[3] Please refer to Google form for the human evaluation results:

https://docs.google.com/forms/u/1/d/1ehFnR-NyAQMTv_j7dvWUA9xT7P2TGRNj0rfiRzoYsIs/edit?usp=sharing

References

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6.

Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. 2016. Coco-stuff: Thing and stuff classes in context. *CoRR*, abs/1612.03716.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915.

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. 2018. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65.

A. Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. 2017. Can: Creative adversarial networks, generating "art" by learning about styles and deviating from style norms. In *ICCC*.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *CoRR*, abs/1508.06576.

Xun Huang and Serge J. Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR*, abs/1703.06868.

Tero Karras, Samuli Laine, and Timo Aila. 2018. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Shervin Minaee, Yuri Y. Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. 2021. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic image synthesis with spatially-adaptive normalization. *CoRR*, abs/1903.07291.

S. Smithsonian Magazine. 2016. An ai-written novella almost won a literary prize.