Style Transfer via Object Detection

Marvin Yang
cyang108@ucsc.edu
University of California, Santa Cruz
Santa Cruz, CA

Kehua Lei
Klei4@ucsc.edu
University of California, Santa Cruz
Santa Cruz, CA

ABSTRACT

Style transfer is a popular topic in machine learning and has numerous developed applications. However, style transfer applications are usually applying the style to the entire image. Here we purpose a novel application that could apply different art styles on different classes of objects and combine those objects into a new artwork.

KEYWORDS

style transfer, object detection, instance segmentation

ACM Reference Format:
Marvin Yang and Kehua Lei. 2018. Style Transfer via Object Detection. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

As machine learning becomes a popular topic, there are more and more machine learning applications been developed. Among all machine learning applications, it could say that style transfer is one of the most popular applications. Style transfer is using the machine learning technique, applies a different style of artwork to an input image, such as applying oil paint art style on photography, making the resulting image have the texture and the tone of oil paint, while the content is base on the input photography.

Numerous style transfer applications have different features, and some of them can even perform real-time video style transfer. Among those style transfer applications, what they do is applying art style on the entire image or video, while we wonder what if we only want to make some specific objects applying style transferring? Such as applying anime art style to the food to make it more delicious but not applying on people, or making the background as a fantasy world while the people stay realistic. With an exciting idea, we develop this application to achieve the goal.

2 METHODS

In this application, we are using two machine learning techniques. The first technique is style transfer. We provide several trained models for different style transfer, allow performing fast style transfer for an image. The second technique is image object detection with segmentation, showing how many classes of objects are detected. We integrate those two techniques into a web-based interface to allow the user to generate their pictures with the desired art style for each class of objects.

2.1 Style Transfer

We take an online project[1] as reference for implementing fast style transfer. This is an implementation of an existing work[4], the network overview is as below.

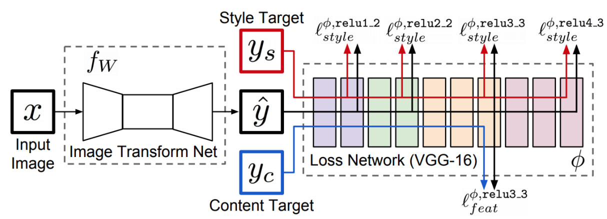

Figure 1: System overview from the referenced paper[4]

Content and style features are computed from VGG19 instead of VGG16. Content Layer is conv4_2 and style layers are conv1_1, conv2_1, conv3_1, conv4_1 and conv5_1. During training, content images are rescaled to 256 * 256, and the shortest side of the style image is rescaled to 512 to reduce the computation. Each model is trained using 40k iteration with learning rate 1e-3, with the COCO dataset 2014, and we trained several models with different art styles.

2.2 Object detection and segmentation

To allow transferring different styles for different classes of objects, we need not only object detection but object segmentation program. We take matterport/Mask_RCNN[2] as a reference to implement object detection and segmentation. This repository is an implementation of an exisiting paper[3]. The model generates bounding boxes and segmentation masks for each instance of an object in the image, with its class and confidence value. The process it completes is first performing object detection then object segmentation. Mask R-CNN is an intuitive extension of Faster R-CNN, by using Region Proposal Network, it proposes candidate object bounding boxes and detects objects' class, while also outputs a binary mask for each Region of Interest in parallel. Below is an example result of Mask R-CNN detecting objects and making object segmentations.

2.3 Web-base Intergration

We designed a web-based application, integrate those two techniques, allowing the user to transfer different classes of objects into different art styles, while combined in the same result image. We

Figure 2: Results generated by models with different art styles

Figure 3: Result of object detection[3]

first load the image from user input, then using Mask R-CNN to perform object detection and segmentation. After we obtain the array of the class of each detected object and the array of the binary segmentation mask of each object, we combine the segmentation mask with the same class.

Next, we display the detected classes and ask the user to choose the desired transforming style for each class. Aside from the detected classes, we also allow the user to choose the transforming style for background. The art style options are below. The user can also choose "none", indicate no art style will be applied and keep original. After choosing the art style for each class, we generate images using fast-style-transform for every selected style, and we extract the image according to the corresponding mask, then combining every extracted image.

3 RESULTS

Using the application, we generate some images for example showing the results.

Some generated images show interesting combinations for putting different styles together. Having a strong contrast art style can make the class of objects become highlighted. Additionally, keeping a class of objects in the original could also highlight the class of objects.

4 DISCUSSION

Using this application to generate images is an interesting experience. In the beginning, we have to observe the style options to guess how to make a good combination, considering the art style, texture, and tone of color. Every time generate an image, there is a sense of novelty observing objects being style transformed, and there is even a higher level of novelty observing those interesting style transformed objects being put together. After having a better sense of what styles can make a good combination, depending on personal preference, the next step is considering how to choose the proper style for each class to generate an interesting result.

Figure 4: Result of segmentation[3]

While most parts of the experience are positive, we did find some elements that could be added.

4.1 Element of surprise!

Although it is interesting to participate in the generating process by deciding the art style of each class of objects, we think a random generator would be a potential feature. It could generate a surprisingly interesting result, sometimes while bringing another sense of novelty.

4.2 Better segmentation

While most of the time, object segmentation is accurate, it does fail to make the precise segment sometimes, and it is frustrating when the class of objects is the desiring highlighting class. Firstly, we can improve the accuracy by using a better-trained model, having more iteration, and using either a larger dataset for general usage or a specific dataset for a specific usage(e.g., animal topic pictures). Secondly, we are considering making it flexible to allow the user to

Figure 5: Final result1

adjust the mask by providing a handy mask-editing tool similar to some image editor feature.

4.3 Smoother style constract

While it is interesting combining and putting different styles of objects together, some styles of objects are having a sharp contrast on texture or tone of color, and we think it could be a useful element to add on if there is a feather option, to make the edges of objects less sharp by blending nearby art style.

Figure 6: Final result2

Figure 7: Final result3

4.4 Efficient pipeline design

In current pipeline, we ask user to upload another image after finishing combination. However, we realize sometimes we are using the same input image because we are going to find out a better combination, and there is no need to go through the whole iteration, such as object detection and regenerate style-transformed images.

REFERENCES

[1] [n.d.]. *Fast style transfer.* Retrieved January 10, 2019 from https://github.com/conan7882/fast-style-transfer

[2] [n.d.]. *Mask R-CNN for Object Detection and Segmentation.* Retrieved March 9, 2019 from https://github.com/matterport/Mask_RCNN

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision.* 2961–2969.

[4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision.* Springer, 694–711.

Figure 9: Final Result4