National Tsing Hua University
Department of Electrical Engineering
EE6620 Computational Photography, Spring 2022

Term Project

Animation Image Harmonization

Group: Team11　Member: 蔡惠芸，涂皓鈞

1. Introduction

Compositing is one of the common usages in photo editing. To composite two images, a foreground object is extracted from one image and pasted on another photo, which called background. However, the appearances of foreground object may not look realistic on the background photo. Therefore, it is important to adjust the color or brightness of foreground to make it consistent with the background photo.

Composite photo can not only be made from real life photo, but also from animation or drawing, which can make a character in one movie appear in another movie scene. Therefore, except for dealing with real world images, we try to make composite animation image harmonious.

2. Motivation

To compare the differences between real-world images and animation images, we observed a dataset, iHarmony4[1], which is commonly used in recent research. We found out that the color of light source of real-world image is not various than animation images. In addition, for animation images, the color saturation is higher than real-world images which make the color and brightness of items, affected by light source, is different from real-world, Therefore, we suppose that the model trained in real-world image dataset cannot perform well in animation images. To deal with this problem, we use animation images as dataset, and refer to survey, we find out a more compatible model structure for animation images.

3. Related Work

Early deep-learning work on image harmonization exploit an encoder-decoder structure with skip connections and an additional branch for semantic segmentation by *Tsai et al* [2]. Encoder-decoder structure is widely used in different computer vision tasks like super-resolution, image denoising and semantic segmentation, etc. Therefore, the other deep-learning works, which is done by *Cun et al* [3] and *Konstantin et al* [4], are also encoder-decoder U-Net-based model. In the leader board provided by iHarmony4 dataset, the U-Net-like model is also leading in good ranking. Therefore, in

our work, we also start from U-Net-based model.

4. Framework

4.1 Dataset construction

Because there is only real-world image harmonization dataset and real-world image segmentation dataset, we prepare composite image dataset and mask image dataset by ourselves refer to iHarmony4.

Animation image selection: We select an animation video consisting of any different short animation video, spilt the video into image frame with 3 frame/sec, and totally get 730 animation images.

Foreground Segmentation: There is no animation image dataset with segmentation mask, so we segment one foreground for each image. To extract the foreground object, we try different python packages to remove background, find out that Rembg tool is the best, so we use Rembg tool to extract the foreground object. (see Fig. 1)

For each image, we ensure that the foreground object in the picture is not occupied 50% above of whole image, to ensure enough background information.

Foreground Adjustment: After extracting the foreground object O_f from one image I, we adjust O_f into other apperance O_d which takes reference from randomly chosen image in the same dataset by color transformation.

We apply two different transformation and randomly choose one transfermation applying to the foreground. The two transformation is pdf transfermation and lab transfermation. Let the input forground object as O_f, reference image as R, $fO_f(r,g,b), fR(r,g,b)$ be the probability density function of O_f and R's RGB value and the output image is O_d.

For pdf transfermation [5], $O_d = t(O_f,)$, it supposes that the transformation *t* is a continous mapping so that $fO_f(r,g,b) = fR(r,g,b)$. The algorithm iterates 300 times for each image, keeps rotating both sample of O_d and R and makes pdf of two images as same as possible.

For lab transfermation [6], we transfer both O_f and R from RGB color into CIELAB color space as O_f' and R', because lightness is more important than human vision and channel between LAB color space is more decorrelated which is more suitable for color transfer. Then we make the mean and standard deviation of O_d' and R' the same by equation 1 and transfer O_d' to O_d from CIELAB color space into RGB color.

$$O_d' = \frac{O_f' - mean(O_f')}{std(O_f')} * std(R') + mean(R') \quad (1)$$

The total preprocessing flow is shown in Fig 1.

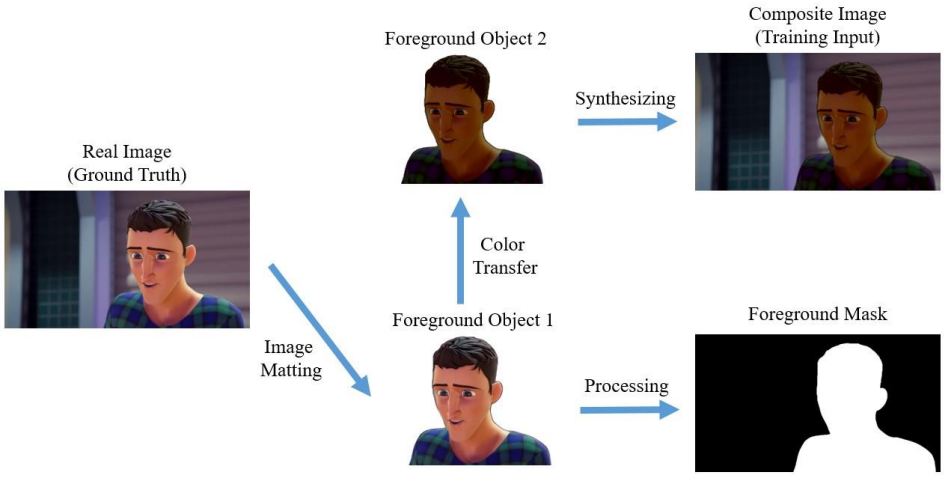

Figure 1: Data preprocessing flow

4.2 Model Architecture

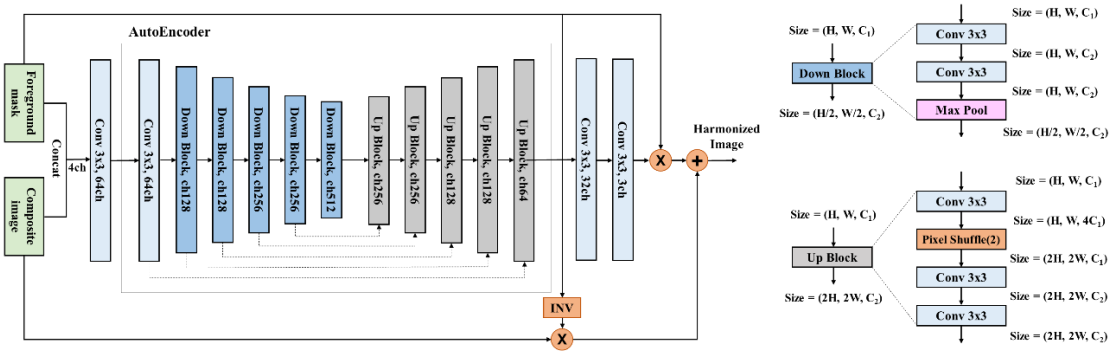

Figure 2: Model architecture

As we mention in Section 3, we also use Unet-based model as our backbone and add a mask at the end of the model output (shown on Fig. 2). Skip connections in auto-encoder structure can help training stable and recover those image details. For other work, they not only predict the harmonized foreground but also semantic segmentation mask thanks for the COCO dataset. It makes the mask more reasonable, and the model can get the semantic segmentation information to generate harmonized image. However, in our task, we only had the foreground segmentation mask data, so we add the mask to the back, making the model directly see the segmentation information.

5. Experiments

Assumption1: In experiment, we found that if we simply put foreground mask into model as showed in the upper of Fig. 3, there are visual artifacts around the edge of the foreground object. There are two reasons for the phenomenon: (1) the effect of image

matting we used is limited in some cases. (2) the pixel value of the mask is extreme (0 or 255), it's hard to adjust by degrees around the edge of the foreground object. Therefore, we assume that the foreground mask with blur can alleviate the artifacts.

Justification1: In the lower of Fig. 3, demonstrating the harmonized image with Gaussian blur on foreground mask. It's obvious to see that the artifacts around the foreground object are alleviated, verifying our assumption.

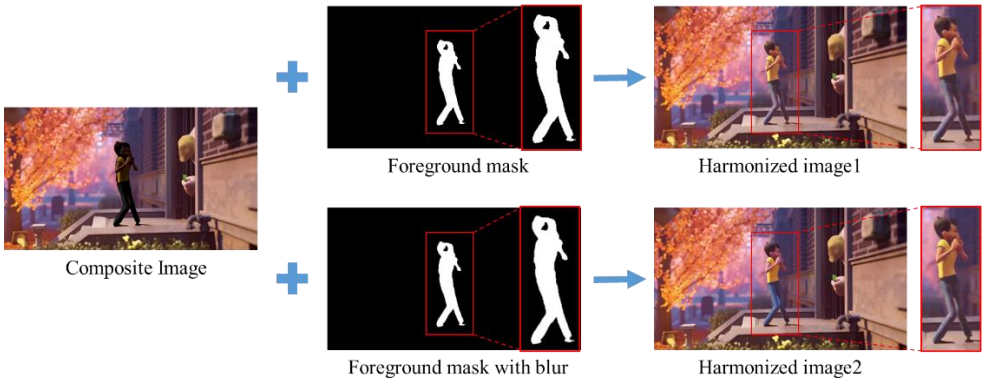

Figure 3: Harmonized images without/with Gaussian blur on foreground mask.

Assumption2: For image harmonization, the foreground object adjusts color along with the background, meaning that the background basically wouldn't be changed. However, when the patch size during training isn't large enough or the foreground objects of testing images are too large, it would degrade the performance. Since it's likely that the random cropped image doesn't contain foreground object during training, specifying enough training patch is necessary.

Justication2: Fig. 4 shows experimental results on different size of patch and foreground objects. It can be seen that there is only half of the foreground object harmonized on small patch size condition. Besides, in the right of Fig. 4 displays that larger foreground object exactly influences the expected performance.

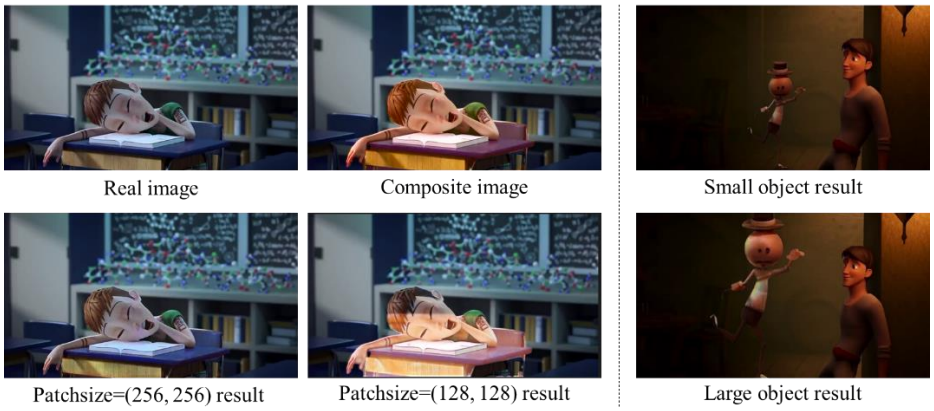

Figure 4: Harmonized results on different size of patch and foreground objects.

Assumption3: As mentioned in Section 2, we suppose that the model trained in real-world image dataset cannot perform well with animation images, since the color of light source and the color saturation are quite different between two datasets.

Justification3: Fig. 5 shows an example result of our work. Note that the composite image was made in handcraft, so there is no ground truth image for comparison, which is more proper for practical applications. Comparing results of ours and iSSAM [4] work, the foreground object seems to be more harmonized with its background. Table 1 records the evaluation results on our dataset. Among the four metrics, fMSE is the sum of the error divided by the number of pixels in foreground, which is commonly used in the application of image harmonization. From the results, it can be verified that the model trained in real-world images perform less well with animation images, which the necessity to perform animation harmonization has been confirmed.

Composite image Ours iSSAM

Figure 5: Example results of ours and iSSAM work on our animation dataset.

	fMSE	MSE	PSNR	SSIM
Ours	**62.17**	**11.83**	**35.83**	**0.9763**
iSSAM	98.59	13.47	34.60	0.9547

Table 1: Evaluation results of ours and iSSAM work on our animation dataset.

6. Conclusion

In this project, we construct our animation dataset in an efficient way to collect large amount of training pairs. Further, we build the deep neural network based on the commonly used Unet-based architecture. In experiment, we conclude that (1) the generated foreground mask with blur helps alleviate the artifacts. (2) the patch size might be large enough during training process and foreground objects cannot be too large to maintain the performance. Finally, in the last part of evaluation, we demonstrate that our model for animation image dataset outperforms the model trained for real-world image dataset, proving that the property of animation and real-world images is quite different. It is practical to establish the model for the specific application — animation image harmonization.

7. Reference

[1] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. *arXiv: Computer Vision and Pattern Recognition*, 2020.

[2] Yi-Hsuan Tsai, Xiaohui Shen, Zhe L. Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2799–2807, 2017.

[3] Xiaodong Cun and Chi-Man Pun. Improving the harmony of the composite image by spatial-separatedg attention module. *IEEE Transactions on Image Processing*, 29:4759–4771, 2020.

[4] Konstantin Sofiiuk, Polina Popenova, Anton Konushin. Foreground-aware Semantic Representations for Image Harmonization. *2021 IEEE Workshop on Applications of Computer Vision (WACV)*

[5] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 2, 3

[6] Franc¸ois Pitie, Anil C Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1-2):123–137, 2007. 3