공학석사학위논문

A Novel Approach to Reduce Visual Gap Using Inpainting GAN

인페인팅 GAN을 이용한 비주얼 갭 감소를 위한 새로운 접근 방법

2023년 2월

서울대학교 대학원

전기 · 정보 공학부

Truong Thanh Hien

A Novel Approach to Reduce Visual Gap Using Inpainting GAN

지도교수 이혁재

이 논문을 공학석사학위논문으로 제출함

2023년 2월

서울대학교 대학원

전기 · 정보 공학부

Truong Thanh Hien

Truong Thanh Hien 의 석사학위논문을 인준함

2023년 2월

위원장 조남익 (인)

부위원장 이혁재 (인)

위원 이태호 (인)

Abstract

Image editing task, more specifically image blending, is a method for image composition to make the composite image looks as natural and realistic as possible. To generate well-blended images, the blending process needs to make the edge of the source images appear seamless and preserve the colors of blending object. However, in the previous works, the recent approaches can only produce realistic blending results without preserving the content of blending region, especially its colors, which is the most important to fashionable photos, or the boundary of blended regions is not seamless enough. Moreover, deep image inpainting methods recently have made impressive progress with advances in image generation and processing algorithms. Based on the above, this study develops a new automatic approach using an inpainting Generative Adversarial Network (GAN) to reduce the domain gap between the source image and the target one. Experiments are conducted for two datasets. Compared to the alternative methods, this method shows that the blending images are not only realistic but the content of the blending region is also preserved. The proposed method is practically simple to carry out while still achieving a comparable efficiency to other state-of-the-art approaches on image composition task.

Keywords: Image Blending, Inpainting, Composite Image, Generative Adversarial Network (GAN), Color Difference Checking (CDC).

Student Number: 2020-27678

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Image blending is an image composition task which aims to blend a certain region from a source image onto a target image. Being one of the most common image editing operations, image blending benefits various applications such as art and entertainment, and data augmentation for several other tasks. For instance, one can change the backgrounds of self portraits and make the generated images more realistic and natural using an image blending technique. This operation can also be used for automatically generating labeled training dataset of classification tasks. However, there exists several issues that make the blended image unrealistic and thus degrade its quality significantly. Specially, such simple cut-and-paste approach with a deep-learning segmentation mask usually results in undesirable artifacts along the object boundary. This problem becomes an obvious challenge in the image blending task. When the foreground with jagged and mixed-with-old-background boundaries is pasted on the new background, there is an abrupt color change between the foreground and background.

Many studies on image blending have been conducted in order to address these boundary artifacts. Alpha blending manually assigns alpha values for boundary pixels, indicating what percentage of the colors are from foreground or background, to smooth the transition between foreground and background. Although it's a simple and quick technique, alpha blending distorts the fine details and introduces ghost effects on the composite photos. Laplacian pyramid blending advocated creating multi-scale Laplacian pyramids for two images and conducting alpha blending at each scale while taking into account multi-scale information. Another strategy tries to enforce gradient-domain smoothness in order to produce smooth border transition. Poisson image blending, which was first proposed to ensure the gradient domain consistency with regard to the foreground image, is the oldest study in this research direction. While Poisson image blending can result in a more realistic result than the alpha blending method, it is very expensive to solve the Poisson equation. Based on the observation that the effectiveness of Poisson image blending seriously depends on the boundary condition, several methods are designed to optimize this dependency. The methods which are based on gradient domain smoothness can smooth the transition between foreground and background. However, the distortion of foreground color and halo artifacts are two of several more undesirable effects that cause significant loss to the foreground content. Fashion image is an example which is highly sensitive to the mentioned loss.

In recent years, many researches are inspired by the combination of traditional image blending methods with a deep-learning network to smooth the boundary. Gaussian-Poisson GAN (GP-GAN) is an approach which combines the strengths of gradient-based constraint to an objective function according to the Gaussian-Poisson equation and generative adversarial networks (GANs). Another work which does not rely on supervised training as GP-GAN proposes a two-stage blending algorithm with a Poisson blending loss and content and style loss from deep features. Although these methods can generate a smooth blending boundary, they still fail to preserve the color of the foreground. Apart from the two above mentioned methods, a new learnable image blending network is proposed. This network directly generates a composited portrait image given a pair of foreground and background images with a seamless boundary. Nevertheless, the network relies on ground-truth composite images obtained by using accurate alpha matte as supervision. Therefore, this work also proposes a mask refinement network to refine the details of the alpha matte mask.

In this study, we propose a framework to reduce the visual gap between foreground and background and make the blended image more realistic. Our method is a new combination of the inpainting task with blending task to generate a natural composite image. An inpainting GAN model can be used not only for image completion, which is to fill in missing regions of one image, but also applied to the image editing task, specifically image blending. Moreover, we propose an algorithm that creates a binary line mask fit to each image with different sizes of the visual gap. Experiments are implemented to show that the proposed approach is effective. The blended images produced by our framework have both a smooth blending boundary and the foreground color preserved. In addition, we take advantage of the inpainting GAN in filling holes, which is the lack from segmentation step, of one image. The results on automatic image blending show that our proposed method outperforms all the baselines and achieves a state-of-the-art performance.

Chapter 2: Related Work

This chapter walks through the concept of image blending, including the previous works. The imperfection of other approaches is the motivation for the proposed method in the next chapters. We also introduce an inpainting method which directly involves in our experiments. We highlight the importance of the image inpainting technique in the creation of blending image. Finally, in the last section, we introduce an automated algorithm to generate the binary mask as a blending helper for our task that works well with various image datasets.

2.1. Image Blending and Techniques

In the first step of the image blending task, the foreground from one image is extracted using image segmentation or matting techniques. Then, this foreground is placed onto another image to form a composite image. Without any further tuning, the edge of the foreground part creates a visually clear boundary which makes the composite image unrealistic. As a common image editing operation, image blending algorithms aim to improve the visual of this boundary between foreground and background.

Image blending techniques commonly falls into two general approaches: traditional methods and deep learning methods. A few popular traditional image blending methods are Alpha blending [1], a simple and fast method using manually selected alpha values, Laplacian pyramid blending [2], which builds multi-scale Laplacian pyramids and alpha blending at each scale, and Poisson image blending [3] that enforces the gradient domain consistency. Applying these methods often introduce several undesirable effects such as color distortion, blurring details, and ghost effects of a blended image. Some others are also slow and inefficient. The latter approach is either inspired by traditional image blending methods or proposed as a new learnable image blending framework. The framework of GP-GAN [4] takes advantages of both GANs and gradient-based image blending methods while Zhang et. al [5] proposed a two-stage deep-learning blending algorithm which does not rely on any training data as GP-GAN. However, both methods change the colors of the blended region. Another approach is a deep-learning-based framework for fully automatic portrait image compositing [6] including foreground segmentation and mask refinement networks. The disadvantage of this method is that it requires ground-truth composite images obtained by using accurate alpha matte as supervision.

2.2. Generative Adversarial Network

Generative Adversarial Networks, or GANs [13], are neural networks used for generative modeling. A generative model generates new samples from a distribution of samples that are similar but specifically different from the existing dataset, such as generating new photographs that are similar but specifically different from an existing dataset. GANs are generative models that are trained using two neural network models. There is a model called a "generator" or "generative network" that generates new plausible samples. It learns to differentiate between generated examples and real examples using the "discriminator" or "discriminative network". Examples of GAN usage includes generating new human poses, inpainting and blending images, and generating examples for image datasets.

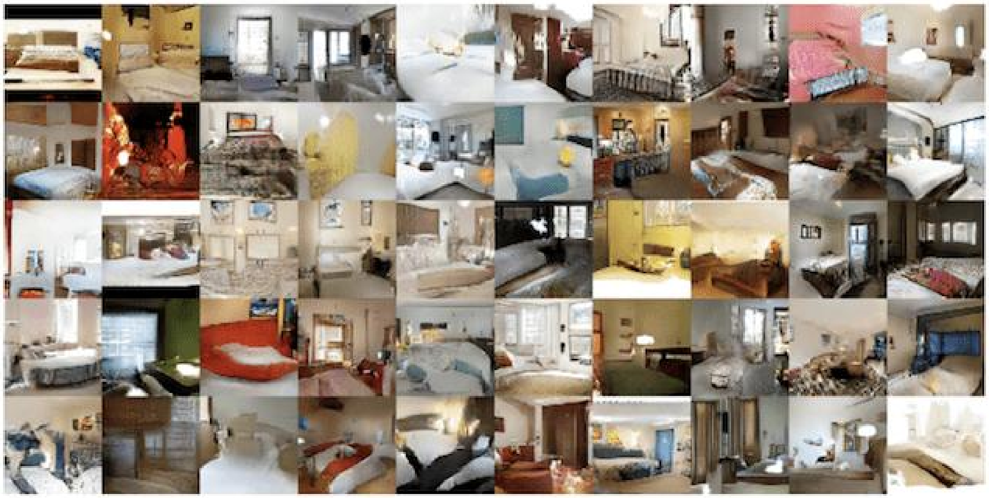


Figure 2.1. Example of GAN-Generated Photographs of Bedrooms [14].

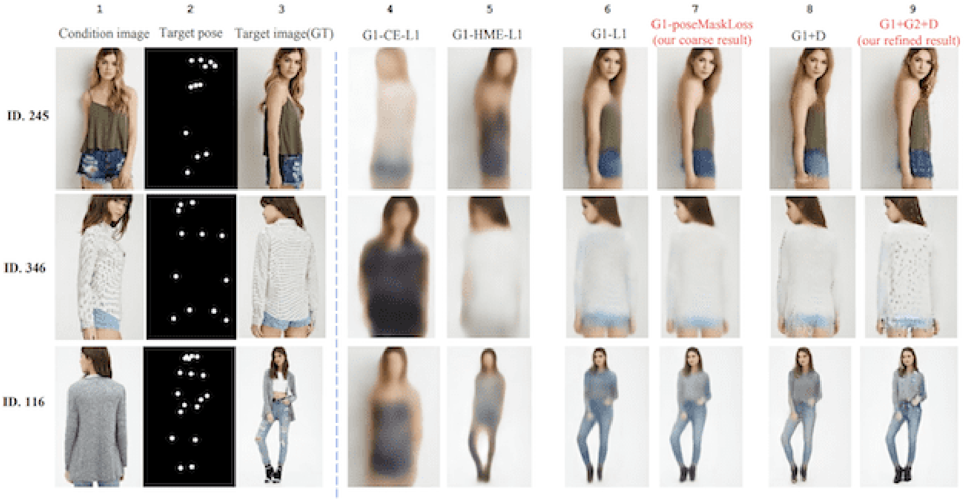


Figure 2.2. Example of GAN-Generated Photographs of Human Poses. Taken from Pose Guided Person Image Generation, 2017 [15].

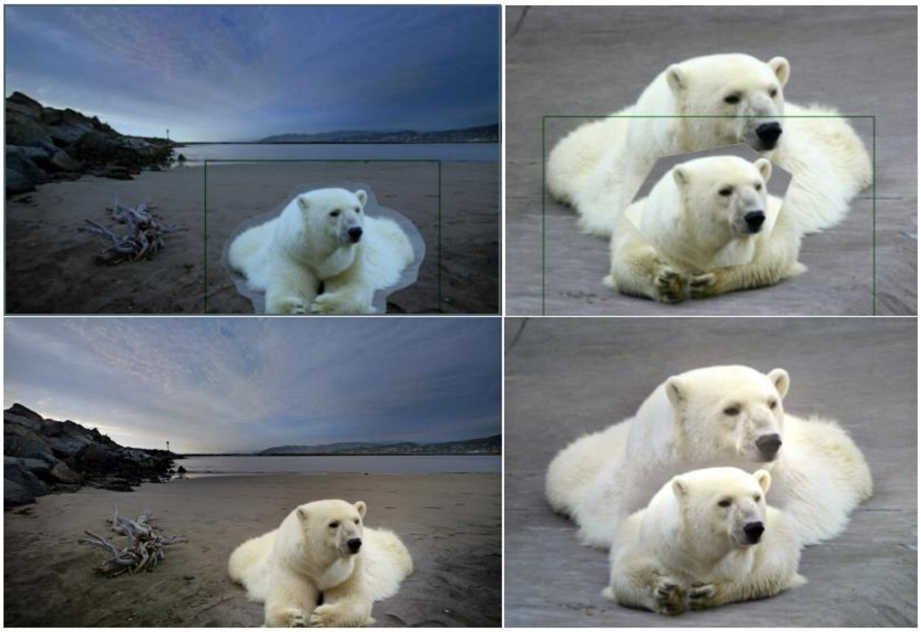


Figure 2.3. Example of GAN-based Image Blending. Taken from GP-GAN: Towards Realistic High-Resolution Image Blending, 2017 [4].

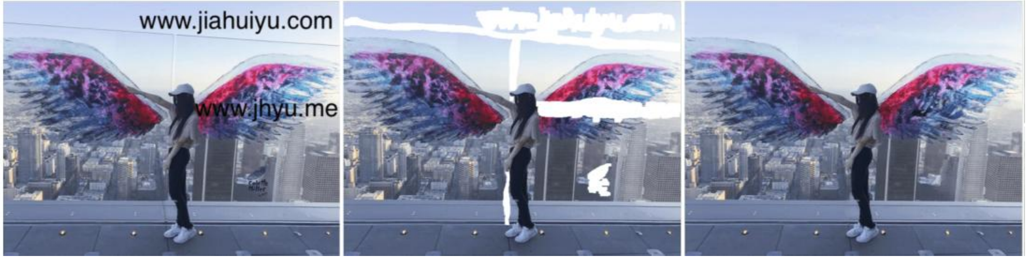


Figure 2.4. Example of GAN-based Image Inpainting [8].

2.3. Image Inpainting

The purpose of image inpainting is to reconstruct missing regions of an image. In Computer Vision, image inpainting has many applications, including image restoration, object removal, compositing, manipulation, re-targeting, and image-based rendering. Additionally, video re-touching, un-cropping, and re-targeting can be accomplished using image painting.

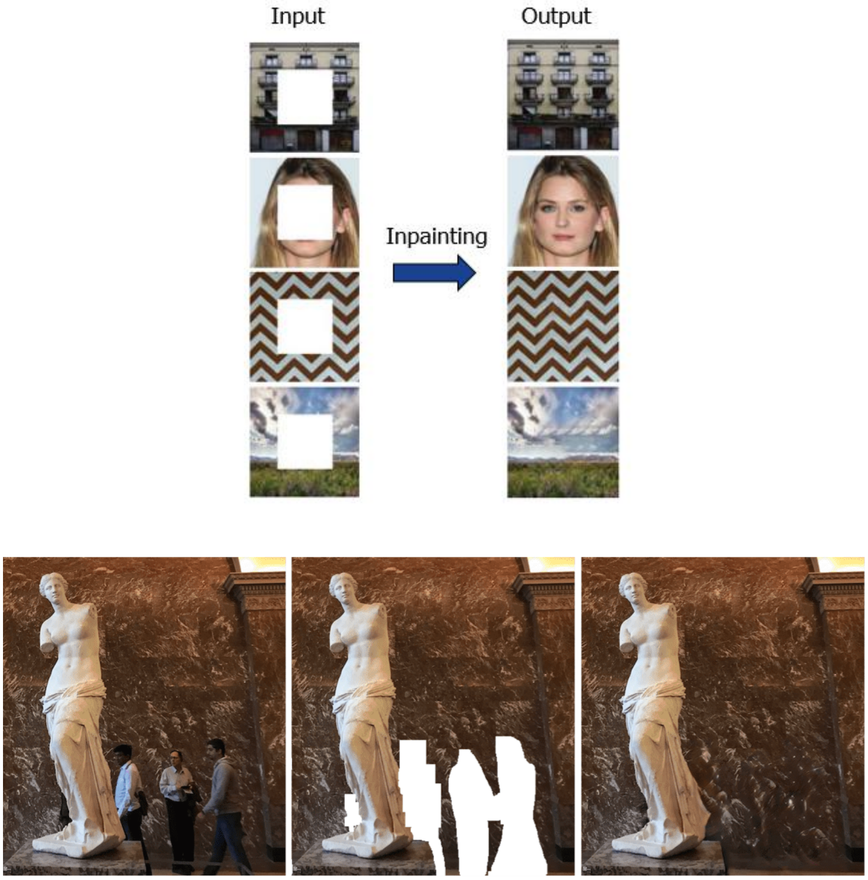


Figure 2.5. Image inpainting examples [8].

Traditionally, inpainting is achieved by borrowing pixels from the surrounding regions of the given image that are not missing. These techniques are good at inpainting backgrounds in an image, but fail to generalize to the cases where the surrounding regions do not have the appropriate information to fill in the missing parts or the missing regions require the inpainting system to infer the properties of would-be-present objects. With the modern approaches, a neural network is trained to predict missing parts of an image. Thanks to the deep learning-based approaches and the era of Big Data, we can now generate the missing pixels in an image with a good global consistency and local fine textures. For inpainting model, an unlimited amount of paired training data can be automatically generated simply by corrupting images deliberately and using the original images before corruption as the ground-truths.

An important challenge in inpainting is that there are many plausible answers for filling in a missing region in natural images, and this ambiguity often leads to blurry or distorted structures. EdgeConnect [9] uses salient edge detection for guiding the inpainting process. Yu et al. proposed DeepFill [8] with contextual attention that refers to surrounding image features to make a better pixel prediction for holes. The deep generative methods [8, 9, 10] based on GAN have shown impressive performance for image completion in recent years.

Chapter 3: Reducing Visual Gap using Inpainting

In the previous chapter, we have discussed the challenge and importance of image blending. There are many blending methods, and each method has its own advantages and disadvantages in terms of efficiency, complexity, and types of artifacts. In this chapter, we propose a method to reduce the visual gap between the foreground and background and to improve the visual quality of the blended image. Our method is the new combination of the inpainting task with blending task to generate more natural composite images.

3.1. The Artifacts and Solution

Color distortion and abrupt intensity change between foreground and background are the two issues that an effective blending technique should fix. We propose a framework that directly address and solve these problems. The overall framework follows the basic procedure of compositing an image from a foreground and a background image. Firstly, we generate the naïve "Copy&Paste" image. Then, a boundary binary mask, named Line Mask image, is generated to define the region that needs adjustment. The Copy&Paste image and Line Mask are combined and fed to an Inpainting network to fill and retouch the boundary of the blended image. The adjustment closes the visual gap at the cutting edge of the two components and makes the results seamlessly blended. Our framework is illustrated in Figure 3.1.

3.2. Preliminary

Given a source raw image x_{raw}, a background image x_{bg} and a binary segmentation mask image x_{mask}, using the naïve copying-and-pasting strategy, a composite image

x_{comp} can be obtained by Equation 1, where $*$ is the element-wise multiplication operator. The operation replaces a region, denoted by the mask image, in the background with the foreground. The goal of the conditional image generation is to generate a well-blended image that is semantically similar to the composite image x_{comp} but looks more realistic and natural at the same resolution.

$$x_{\text{comp}} = x_{\text{raw}} * x_{\text{mask}} + x_{\text{bg}} * (1 - x_{\text{mask}}) \tag{1}$$

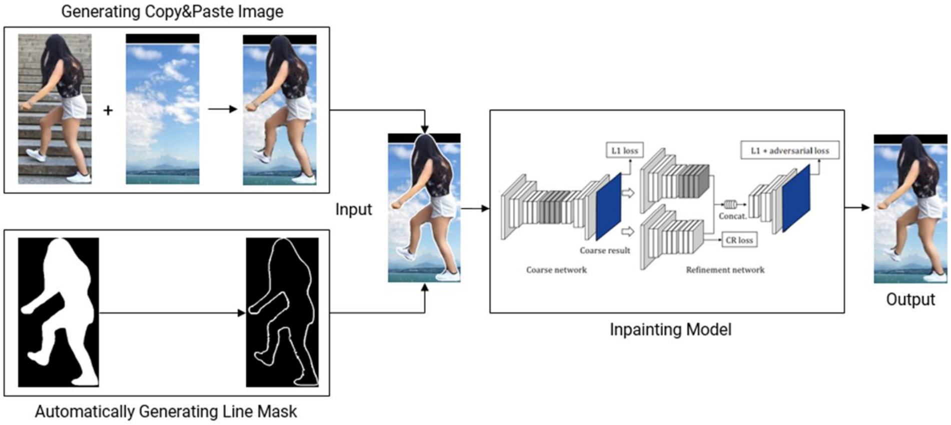


Figure 3.1. The proposed framework.

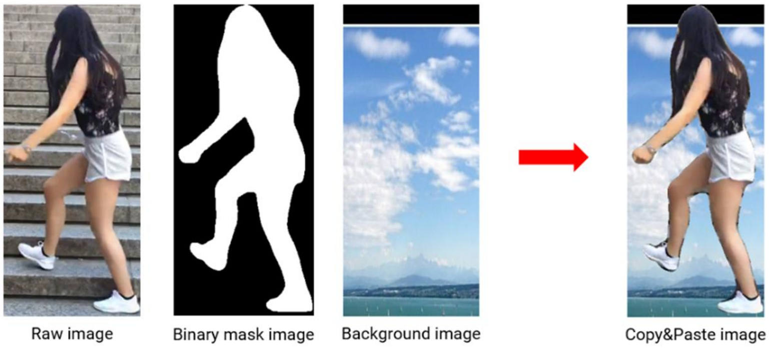


Figure 3.2. The stage of generating Copy&Paste image.

3.3. Introduction of Line Mask

The Segmentation Mask is not perfect. There are points that belong to the background included in the mask and there are points that belong to the object excluded from the mask as illustrated in Figure 3.3. In order to blend the object image appropriately, we propose to finetune the mask with another mask, named Line Mask. Line Mask is a line drawn at the object contours that varies in thickness at different points. The purpose of Line Mask is to identify the noise pixels (pixels that should not belong to the Segmentation Mask). Later during the blending procedure, those pixels will be replaced to improve the visual quality. Line Mask is allowed to include pixels that belong to the actual object as well, but only those at the cutting edge of the object. Blending algorithm can alter the value of those pixels to fit them with the new background.

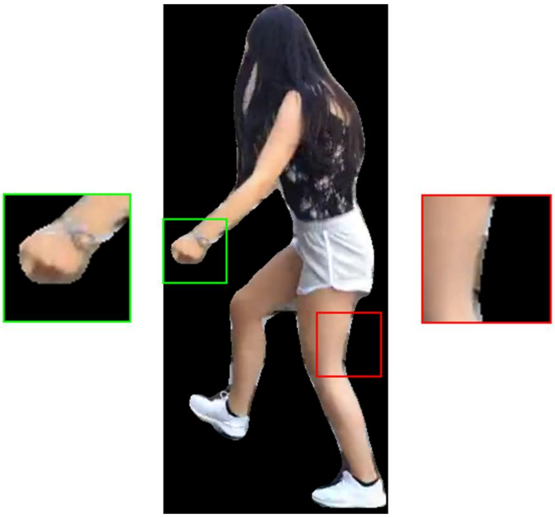


Figure 3.3. Object isolated by a segmentation mask (middle). Several pixels that belong to the background are included as shown in the right sample. On the other hand, the fingertip of the person is incomplete as shown on the left sample.

We propose using a Color Difference Checking (CDC) algorithm to draw the Line Mask. The algorithm is applied mostly to the pixels at the contours of the isolated object. CDC utilizes the difference of color to classify which pixels are noise or not, under the assumption that the Segmentation Mask is able to fit measurably tightly to the object. The algorithm has several hyper-parameters that we optimize based on the performance of the segmentation model and the theory of color differentiation.

3.4. Color Difference Checking algorithm

Color Difference Checking (CDC) algorithm is written generally in Algorithm 1. The algorithm uses color value provided by the original image and an initial classification of object and background points from the binary mask. It considers a list of points, which in our case, the contours of the isolated object. As mentioned previously, those points need further consideration to separate them into the actual object and actual noise with a decent accuracy for the generation of Line Mask, which we will discuss in the next section.

The algorithm in general is iterative comparison. For each point $\boldsymbol{p}$ to be classified, we compare its color value to that of a reference point $\boldsymbol{q}$, which is $\boldsymbol{k}$ pixel away following provided direction $\boldsymbol{di}$ from it (for example, if the direction $\boldsymbol{di}$ is left, $\boldsymbol{q}$ is to the left of $\boldsymbol{p}$ and x-coordinate of $\boldsymbol{p}$ is $\boldsymbol{k} - 1$ higher than that of $\boldsymbol{q}$). It is worthy to clarify that the distance $\boldsymbol{k}$ is calculated, including two end points $\boldsymbol{p}$ and $\boldsymbol{q}$. The distance $\boldsymbol{k}$ starts at 2, being the first comparison of $\boldsymbol{p}$ with its neighbor pixel. Until $\boldsymbol{k}$ reaches its maximum allowed value, we keep changing the reference point and comparing the colors. When there is a reference point $\boldsymbol{q}$ that is actually different from $\boldsymbol{p}$ following our criteria, no more checking of $\boldsymbol{p}$ is necessary. In this case, we can say that $\boldsymbol{p}$ is a noise pixel, and other pixels from ***p*** to right before ***q*** in the selected direction are also noise. Otherwise, ***p*** is considered an actual object point. All the results are tracked, including the query point ***p*** and the distance ***k***.

The algorithm can be repeated with each of four directions: left, right, top, and bottom. With the gradual increment of ***k***, non-convex Segmentation Mask is no problem since the algorithm can always detect an invalid reference point and early stop. Two examples of iterative comparison are illustrated in Figure 3.4 and 3.5. In Figure 3.3, the algorithm is allowed to reach up to ***k*** = 5, but it will likely stop at ***k*** = 3 as the threshold has been surpassed. If the direction is right, the algorithm may be at the risk of checking the invalid reference points due to non-convexity. However, it is actually safe as it stops at ***k*** = 2, after finding out that the reference point is not a part of the object. This information is provided by the Segmentation Mask.

The selection of the threshold $\mathbf{T_d}$ is based on Weber's Law of Just Noticeable Differences. The theory states that "two stimuli must differ at a minimum percentage to be perceived as different". In case of light and color, the percentage is proved to be 8%. In addition, the range of intensity for using 8-bit color is from 0 to 255. Following Equation 2, color-difference threshold can be determined to be 255 × 8% = 20. Therefore, we select this value as the CDC algorithm threshold.

$$\text{minimum perceivable percentage} = \frac{\text{threshold}}{\text{maximum intensity}} \qquad (2)$$

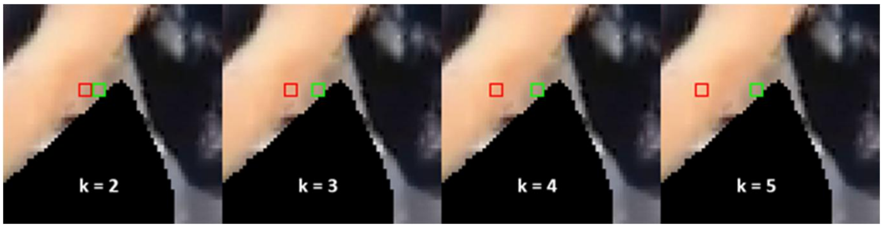


Figure 3.4. Illustration of iterative comparison in CDC algorithm. The selected direction is left. Query point p is marked green and reference points q are marked red. Reference points are selected by the distance k. In a real scenario, the algorithm may early stop at k = 3 if the color difference surpasses the threshold.

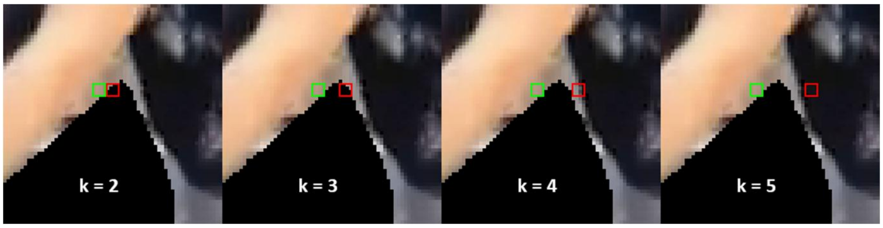


Figure 3.5. Illustration of iterative comparison in CDC algorithm. The selected direction is right. Query point p is marked green and reference points q are marked red. Reference points are selected by the distance k. In a real scenario, the algorithm never reaches k = 3 or 4 or 5 because the reference point at k = 2 is already invalid. It shows that CDC is safe for non-convex masks.

Algorithm 1: Color Difference Checking algorithm

```
Input: Image Im ($h \times w$), binary Mask M, list of points to check C, checking
direction $di$, color-difference threshold $T_d$, maximum checking distance $k_{max}$
/* $di$ can be left, right, top, or bottom */
initialize an empty list of noise points $C_{noise}$
initialize an empty track of thickness $TH_{noise}$ for each point in $C_{noise}$
for $p$ in C do
    /* with each point $p$ in the list to check, C */
    $k = 2$
    while $k \leq k_{max}$ do
        find point $q$ whose distance in the direction $di$ from $p$ is $k$
        if $q$ not exists then
            /* already reached the border of the image; nothing else to check */
            break
        if $M[q] == 1$ then
            /* $q$ exists and belongs to the object part (value of mask at $q$ is 1) */
            $d = | Im[p] – Im[q] |$ // absolute color difference between $p$ and $q$
            if $d > T_d$ then
                /* $p$ and other $k$ - 1 pixels following direction di are noise */
                add $p$ to $C_{noise}$ and $k$ to $TH_{noise}$ // track $p$ and $k$
                break
            $k = k + 1$
            /* if the difference is not much, increase the checking depth $k$ */
Output: $C_{noise}$ and $TH_{noise}$
```

3.5. Generation of Line Mask

We generate Line Mask as a tuning of Segmentation Mask. There are three phases to tune this mask. All three phases have a similar procedure using Color Difference Checking (CDC) algorithm but different purposes. The first phase identifies object points within the contours of the initial isolated object found by the Segmentation Mask. From those points, the second phase finds which points that initially classified as background should actually belong to the object region. It is worthy to remind that CDC can tell not only which point is noise or not, but also how many points near it are. In the implementation, the second phase uses the inverted Segmentation Mask. After the second phase, it is expected that the original Segmentation Mask is expanded (the object region should grow larger and include more points). A new mask that contains new object points found in the second phase is now considered in the third phase. We once again use CDC to classify points, tune the contours of the new mask, and track the thickness of noise. Finally, at the post-processing, we smooth out the thickness map with Gaussian Blur and thresholding. An example of results after each phase is illustrated in Figure 3.6.

Algorithm 2: Line Mask generation

```
Input: Image Im ($h \times w$), binary Mask M, checking direction $di$, color-difference threshold $T_d$
/* $di$ can be left, right, top, or bottom */
calculate maximum checking depth $k_{max}$ based on $h$ // YOLACT-550: 1% of $h$
/* phase 1: find the object points */
find contours C of M
$C_{noise\ phase\ 1}$, $TH_{noise\ phase\ 1}$ = CDC(Im, M, C, $di$, $T_d$, $k_{max}$)
/* repeat with 4 different $di$ */
$C_{obj\ phase\ 1} = C - C_{noise\ phase\ 1}$
/* phase 2: find refined mask $M_{refined}$ that includes missing object points of M */
$M_{inverted} = 1 - M$ // invert the Segmentation Mask
$C_{obj\ phase\ 2}$, $TH_{obj\ phase\ 2}$ = CDC(Im, $M_{inverted}$, $C_{obj\ phase\ 1}$, $di$, $T_d$, $k_{max}$)
/* also repeat with 4 different $di$ */
/* phase 3: find refined mask $M_{refined}$ that includes missing object points of M */
generate $M_{refined}$ from $TH_{obj\ phase\ 2}$
find contours $C_{refined}$ of $M_{refined}$
$C_{noise\ phase\ 3}$, $TH_{noise\ phase\ 3}$ = CDC(Im, $M_{refined}$, $C_{refined}$, $di$, $T_d$, $k_{max}$) )
/* also repeat with 4 different $di$ */
/* post-processing: generate and smooth out Line Mask */
generate Line Mask $M_{Line}$ from $TH_{obj\ phase\ 2}$
smooth out $M_{Line}$ with Gaussian Blur
binarize $M_{Line}$
Output: $M_{Line}$
```

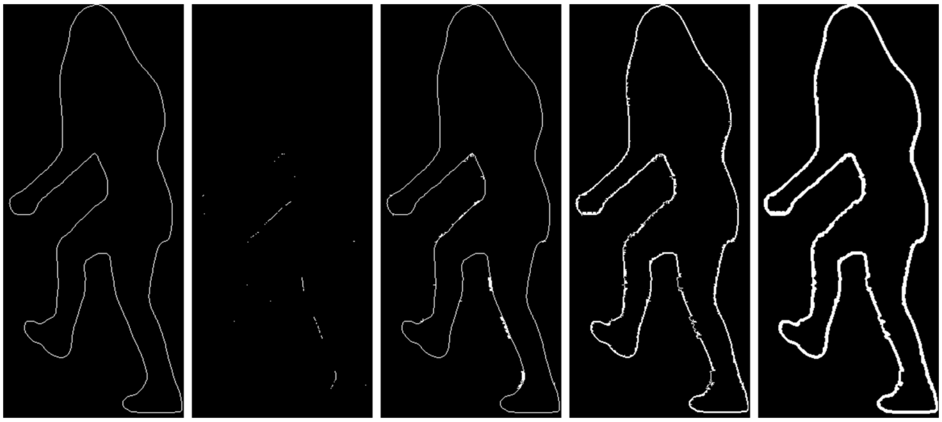


Figure 3.6. From left to right: (1) contours of the original Segmentation Mask, (2) object points found after phase 1, (3) phase 2 thickness map, (4) phase 3 thickness map, and (5) Line Mask after smoothing out phase 3 result.

The summary of Line Mask generation algorithm is shown in Algorithm 2. While phase 1 and 3 utilize CDC algorithm as its core, phase 2 is a little different in the implementation. Instead of using the Segmentation Mask, this phase uses its inverted version in which now 0 actually denotes an object region and 1 denotes a background region. It is necessary to do so to appropriately utilize the implementation of CDC. In addition, CDC is always repeated with four different directions. The output at each phase is the output after all four iterations. Last but not least, as we use YOLACT-550 [11] to generate the Segmentation Mask, the maximum distance kmax is determined as 1% of the image height. This calculation can be derived based statistics, considering the performance of YOLACT-550 having an average of 90% Intersection-over-Union (IoU).

3.6. Image Inpainting Model

Following the success of image inpainting, we propose an idea to take advantage of an inpainting model to generate well-blended images. We introduce the inpainting model, CR-FILL [10], which is used as the tool of our framework to give the realistic composite image. Figure 3.7 is the stage of using network CR-FILL's generator network to obtain results. The coarse network takes an incomplete image where missing pixels are set to zero and a line mask indicating the missing region as input and generates an initial prediction. Then the refinement network takes this initial prediction as input and outputs the final inpainting result.

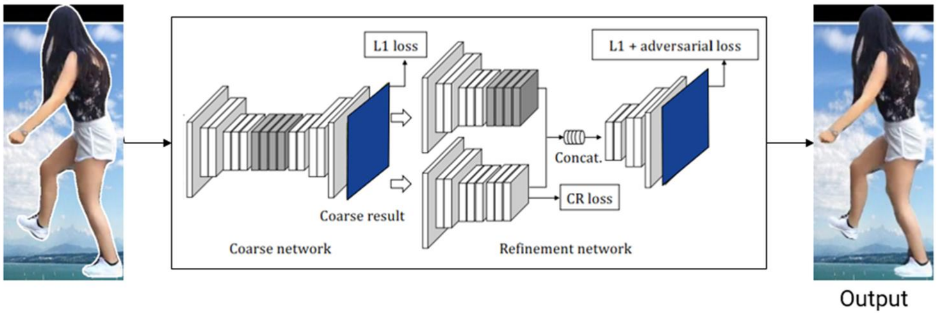


Figure 3.7. Overall architecture of the CR-FILL generator network [10].

Chapter 4: Experimental Results

This chapter shows the experimental results of the proposed method. We perform various experiments with different metrics to prove the advantages of using our method. In each experiment, our method is compared with several baseline methods on different image datasets.

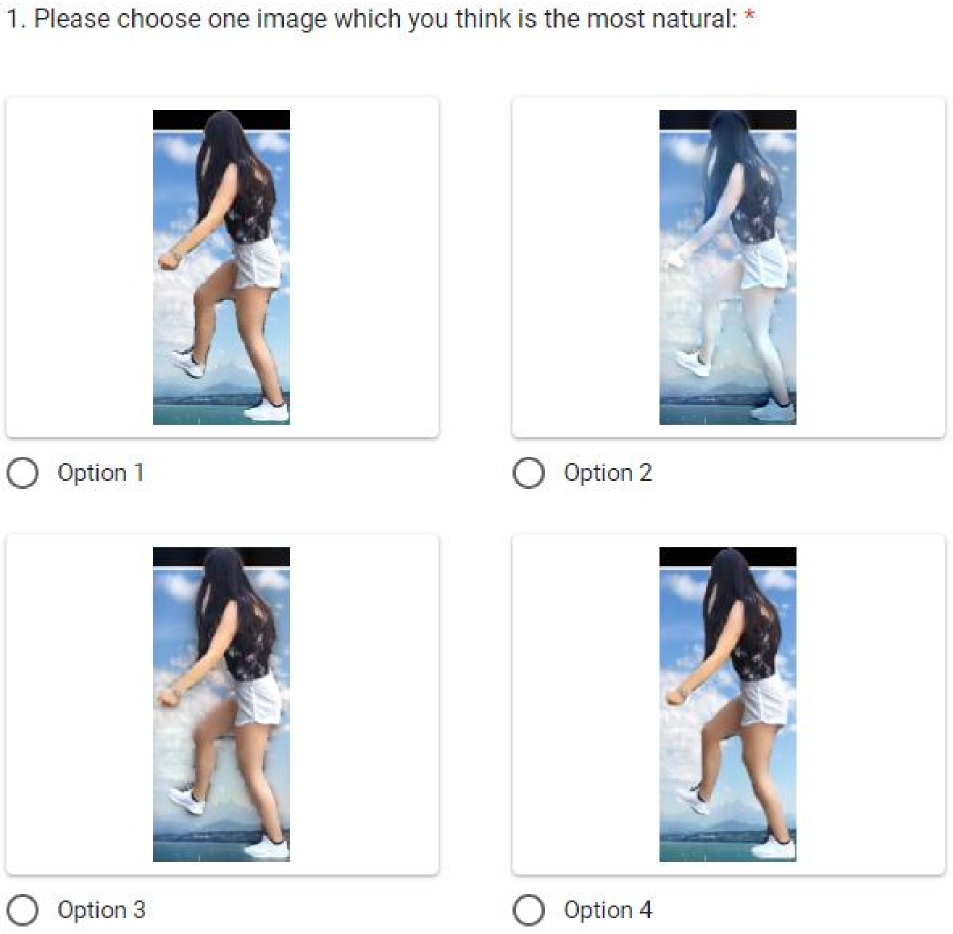
4.1. TikTok Dataset

To evaluate our framework, we blend images from Tiktok dataset [12] with 2615 human images. We also use the YOLACT-550 segmentation network on images in this dataset to get the segmentation mask, which is then used to extract the human images as the foreground. We compare our results with the ones obtained using several intuitive and strong baselines.

The naïve method, named Copy&Paste, produces results that have visually obvious artificial boundary because there is no adjustment added to the blended image. GP-GAN is able to produce a smooth blending boundary. However, color distortion between the blending region and the background is also introduced. The enhanced version of GP-GAN, Combined GP-GAN, makes the boundary seamless in most cases. However, color distortion still exists if the foreground and background image have two different color tones.

The quantitative evaluation uses the standard metrics Peak Signal-to-Noise Ratio (PSNR) to demonstrate the compositing quality, and it serves as a verification process. Another metric we used is the Structural Similarity Index Measure (SSIM). These metrics require a ground-truth image with the segmentation mask provided by the author of dataset. We evaluate the difference between that ground-truth perfect-cut foreground and the same region after applying the blending retouch

Figure 4.4 illustrates the visual difference among all methods. The detailed results are shown in the Table I. The Copy&Paste method introduces no additional adjustment. Therefore, its blended results have the color of the original foreground well preserved that is proved by having higher PSRN and SSIM scores than GP-GAN and Combined GP-GAN. Our method retouches only the cutting edge, which is marked by the line masks, of the foreground and background, thus, there is almost no color distortion. However, the proposed method has even higher PSNR and SSIM, 72.8867 and 0.9321, than the Copy&Paste method, 72.3288 and 0.9280. That means the method is able to fill and adjust the boundary with an appropriate texture while also improving the visual quality of the blended image. This is the concrete proof that the proposed method is superior than others and becomes a new state-of-the-art for the blending task.



1. Please choose one image which you think is the most natural: *

○ Option 1

○ Option 2

○ Option 3

○ Option 4

Figure 4.1. User study survey form. Sample images are shuffled.

Table I. PSNR and SSIM results for our method and the baselines (higher is better). The best scores are in bold.

Method	SSIM	PSNR (dB)
Copy&Paste with Deep-Learning Mask	0.9280	72.3288
GP-GAN	0.8601	65.1327
Combined GP-GAN	0.9115	69.6976
Proposed method	**0.9321**	**72.8867**

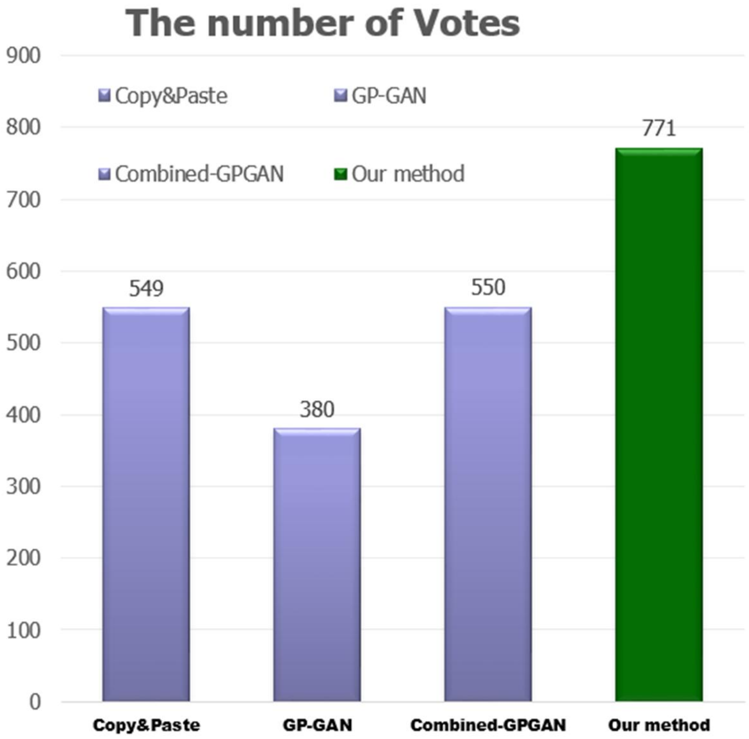


Figure 4.2. User study results. From left to right: (1) Copy&Paste with Deep-learning Mask, (2) GP-GAN, (3) Combined-GPGAN, (4) Proposed method.

For this dataset, we also perform a survey and obtain opinions from forty-five users in order to quantify the performance of all the methods. Each subject is asked to pick one blended image out of four generated by four algorithms which they find to be the most realistic. The survey form is illustrated in Figure 4.1, and the result is shown in Figure 4.2. It is evident that our proposed method can produce the quantitatively best results as the number of votes on it are the highest among all.

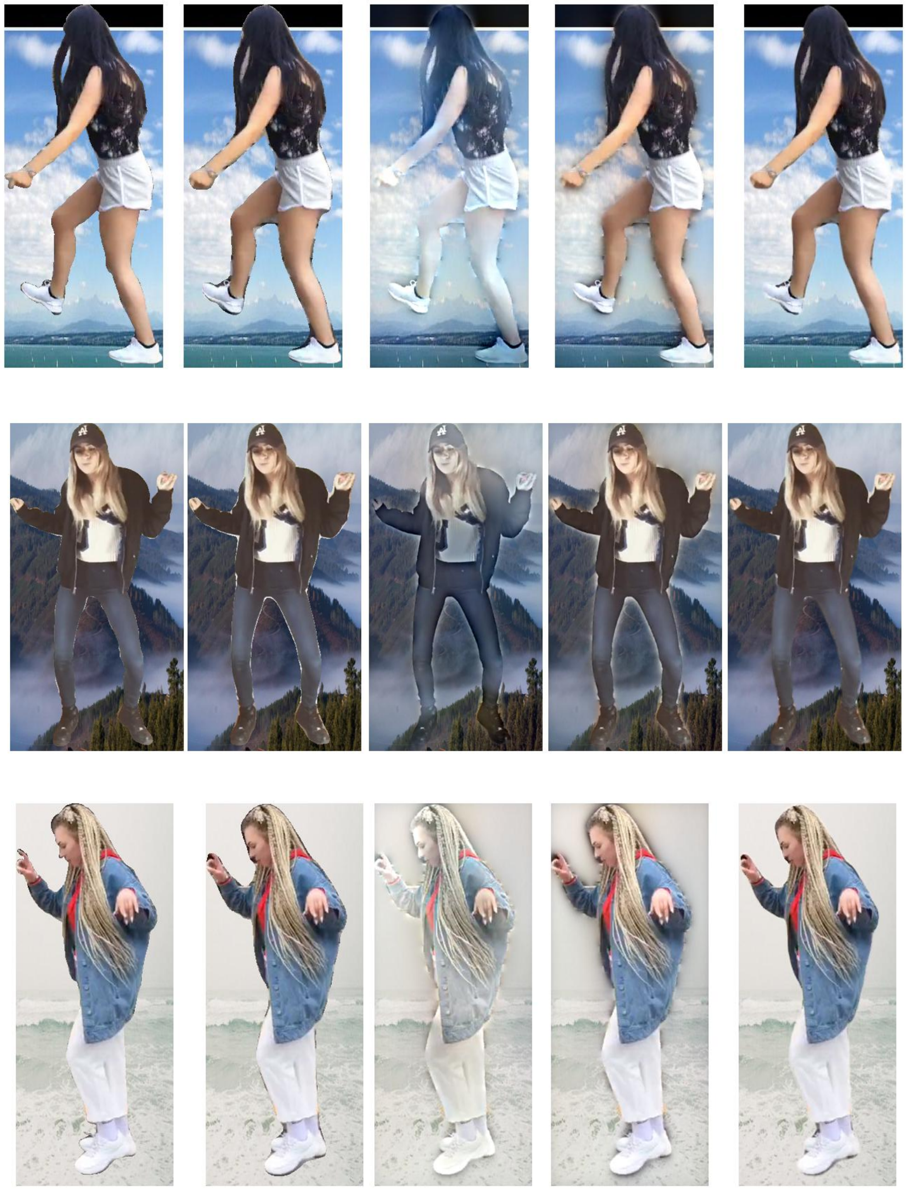


Figure 4.3. The comparison between our method and other baseline methods. From left to right, (1) ground-truth, (2) Copy&Paste with Deep-Learning Mask, (3) GP-GAN, (4) Combined GP-GAN, and (5) our proposed method.

4.2. Fashion Dataset

Following the good performance on Tiktok dataset, we verify our proposed method on another fashion image dataset with 416 model photos onto 30 different background images. The mask images are also obtained using the YOLACT-550 segmentation model on the raw images. We keep on comparing our approach to GP-GAN, Combined GP-GAN, and Copy&Paste.

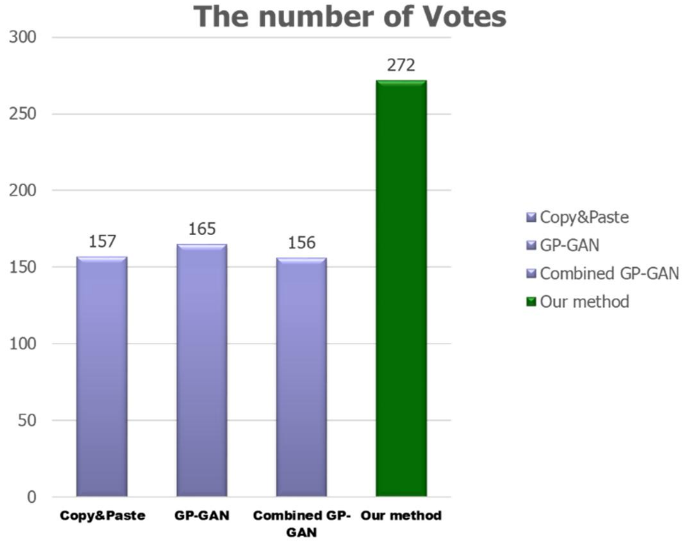


Figure 4.4. User study results.

Because this dataset has no ground-truth binary mask and image blending lacks good quantitative evaluation metrics. Therefore, to quantify the performance of all the methods for this dataset, we perform a survey and collect opinions from twenty-five users for studies. Each subject is asked to pick one blended image out of four generated by four algorithms that they find to be the most natural and realistic. With the results shown in Figure 4.3 and Figure 4.4, it is clear that our proposed method outperforms current state-of-the-art techniques quantitatively and qualitatively.

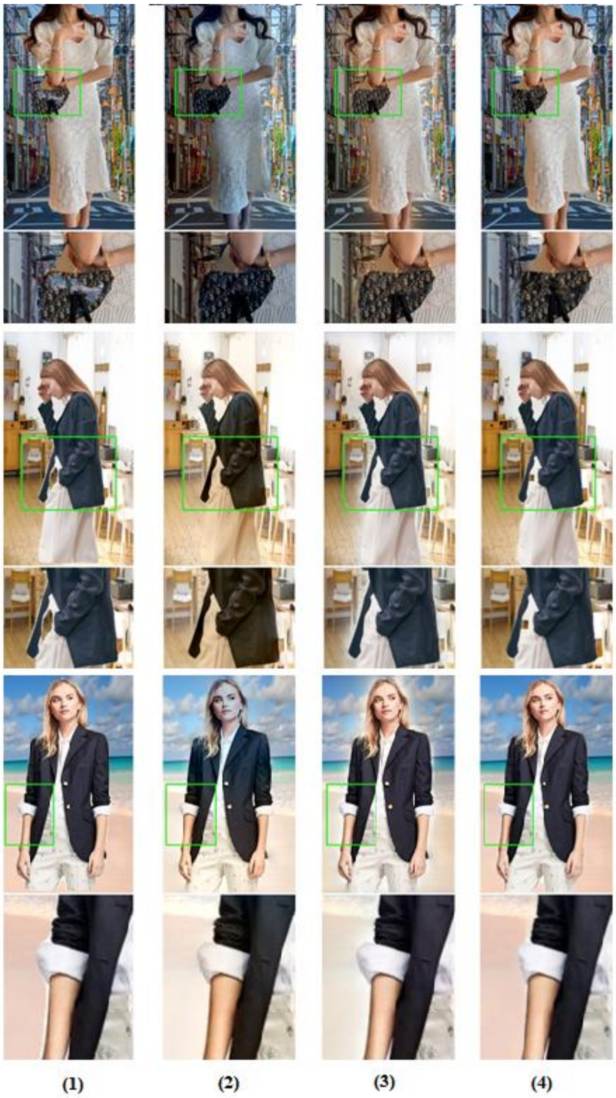


Figure 4.5. Results compared with other methods. From left to right, (1) Copy&Paste, (2) GP-GAN, (3) Combined GP-GAN, and (4) proposed method.

Chapter 5: Conclusion

In this work, we propose a novel image blending framework using Inpainting to generate realistic and natural images without introducing artifacts or color distortion. The method follows a basic stage of image composition from a foreground and a background image, with the blending region marked by a binary mask. An algorithm is proposed to generate a line mask at the boundary of the foreground and background. The line mask redefines the cutting edge of the original mask by including relevant pixels and excluding some others for adjustment. We use the CR-FILL generator network to inpaint the region marked by the line mask to blend two images seamlessly. Since there is no adjustment to the main foreground texture, our method successfully preserves its color scheme, which is important for some applications such as blending human model photos for fashion images. The effectiveness of our proposed method is proved via a user study as well as quantitative experiments. The results demonstrate that our blended images are most voted by users for the highest visual quality among four different methods. PSNR and SSIM scores of our results are also higher than others, setting a new state-of-the-art of the blending task using inpainting. Last but not least, our method is simple, yet efficient to carry out in practice.

Reference

[1] Porter, Thomas, and Tom Duff. "Compositing digital images." In Proceedings of the 11th annual conference on Computer graphics and interactive techniques, pp. 253-259. 1984.

[2] Burt, Peter J., and Edward H. Adelson. "A multiresolution spline with application to image mosaics." ACM Transactions on Graphics (TOG) 2, no. 4 (1983): 217-236.

[3] Pérez, Patrick, Michel Gangnet, and Andrew Blake. "Poisson image editing." In ACM SIGGRAPH 2003 Papers, pp. 313-318. 2003.

[4] Wu, Huikai, Shuai Zheng, Junge Zhang, and Kaiqi Huang. "Gp-gan: Towards realistic high-resolution image blending." In Proceedings of the 27th ACM international conference on multimedia, pp. 2487-2495. 2019.

[5] Zhang, Lingzhi, Tarmily Wen, and Jianbo Shi. "Deep image blending." In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 231-240. 2020.

[6] Zhang, He, Jianming Zhang, Federico Perazzi, Zhe Lin, and Vishal M. Patel. "Deep image compositing." In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 365-374. 2021.

[7] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

[8] Yu, Jiahui, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang.

"Free-form image inpainting with gated convolution." In Proceedings of the IEEE/CVF international conference on computer vision, pp. 4471-4480. 2019.

[9] Nazeri, Kamyar, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi. "Edgeconnect: Generative image inpainting with adversarial edge learning." arXiv preprint arXiv:1901.00212 (2019).

[10] Zeng, Yu, Zhe Lin, Huchuan Lu, and Vishal M. Patel. "Cr-fill: Generative image inpainting with auxiliary contextual reconstruction." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14164-14173. 2021.

[11] Bolya, Daniel, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. "Yolact: Real-time instance segmentation." In Proceedings of the IEEE/CVF international conference on computer vision, pp. 9157-9166. 2019.

[12] Segmentation Full Body TikTok Dancing Dataset. URL: https://www.kaggle.com/datasets/tapakah68/segmentation-full-body-tiktok-dancing-dataset

[13] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks." Communications of the ACM 63, no. 11 (2020): 139-144.

[14] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

[15] Ma, Liqian, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. "Pose guided person image generation." Advances in neural information

processing systems 30 (2017).

초 록

이미지 편집 작업에서 이미지 블렌딩은 합성 이미지를 최대한 자연스럽고 사실적으로 보이게 하기 위한 이미지 합성 방법입니다. 잘 혼합된 합성 이미지를 생성하려면 혼합 프로세스가 소스 이미지의 가장자리를 원할하게 표시하고 혼합 개체의 색상을 잘보존해야 합니다. 그러나 이전 연구에서 최근의 접근 방식은 블렌딩 영역의 내용, 특히 패셔너블한 사진에서 가장 중요한 요소인 자체의 색상을 보존하지 않고 현실적인 블렌딩 결과만 생성할 수 있거나 블렌딩 영역의 경계가 충분하지 않습니다. 더욱이, 딥 이미지 인페인팅 방법은 최근 이미지 생성 및 처리 알고리즘의 발전과 함께 인상적인 진전을 이루었습니다. 위의 내용을 바탕으로 본 연구에서는 소스 이미지와 대상 이미지 사이의 도메인 격차를 줄이기 위해 인페인팅 생성적 적대적 네트워크 (GAN)를 이용한 새로운 자동 접근법을 개발한습니다. 실험은 두 개의 데이터 세트에 대해 수행 하였습니다. 이 방법은 기존의 방법과 비교하여 혼합 이미지가 사실적일 뿐만 아니라 혼합 영역의 내용도 잘 보존된다는 것을 보여주었습니다. 제안 방법은 이미지 구성 작업에 대한 다른 최첨단 접근 방식과 유사한 효율성을 달성하면서 실질적으로 수행하기에 편리합니다.

주요어: Image Blending, Inpainting, Composite Image, Generative Adversarial Network (GAN), Color Difference Checking (CDC).

학번: 2020-27678

Acknowledgement

I would like to express my sincere gratitude to my advisor, Prof. Lee Hyuk Jae for his kind and continuous support of my study and research.

From the deepest of my heart, I would like to say thank you to my family, my husband and my friends for the encouragement, understanding and sympathy.

I also want to thank all members of CAPP Lab. The working environment that CAPP Lab provided fits me well, and the support from my fellows are just great.

Last but not least, I would like to thank my thesis reviewers, Prof. Lee Tae Ho and Prof. Choo Nam Ik, for giving me valuable comments on my research.