CMPT 361 Assignment1

Hanra Jeong (301449735)

Part1: Green screen matting for Virtual Tourism

1.0 Prepare Data:

Portrait_blue: level 1 (neat hair)	Portrait_blue: level 2 (messy hair)	Portrait_blue: level 3 (More dynamic posture, more shapes)	Portrait_green: level 1 (neat hair)	Portrait_green: level 2 (messy hair)	Portrait_green: level 3 (Raised body, more shapes)
			These photos are used for Triangulation of improved algorithm		
Portrait_blue: background picture without foreground image		Portrait_green: background picture without foreground image	3 photos of tourist attraction: Neuschwanstein Castle in German with different angles and field of view		

Figure 1. Prepared data: Images used for the assignments

* Untied hair causing the messy outlines of foreground portrait image causing the increase in level.
* Having more dynamic posture caused more complicated outline of foreground portrait and caused complicated shadow appeared on the wall.
* The used background images were downloaded from [9] to [11].
* Any external software, like Photoshop is not used. The background color blue was color of the wall and green was the color of the paper. These pictures were taken after manually changed the background color with the help of my friend.

1.1 Basic algorithm:

The basic algorithm is for separating the foreground by binary alpha map. This is done by using the hue value by analyzing the hsv of image.[5] By comparing the h values of original images with the background image that doesn't have any foreground, the mask is made if the difference between these two values is less than 3%. By doing this, I could have eliminated the noise and improve the computed images' quality.

To be more specific with the algorithm, the blue color hue value usually ranges from 0.5 to 0.6667 [5], I tried to allocate the mask for the blue range hue values. However, since some of the foreground images also have the pixels that have the hue values in this range, it caused the noise of the mask. Therefore, in order to increase the quality, I allocated the value of 1 for the mask, and converted it to 0 later by *imcomplement()* function, where the

difference between hue values of original image and the background image is less than 3%. The pixel which has less than 3% difference of h values are reliable to said it has the very similar or same color with the background, so meaning it is more likely to be background.

```
for i = 1:a
        for j = 1:b
            if(abs((hsv_h(i,j)-bg_h(i,j))/bg_h(i,j)) < 0.03)
                mask(i,j) = 1;
            end
        end
    end
```

Code 1. Comparing the h values of background image with the original image

To minimize the noise for the computed mask, *imerode()* and *imfill()* functions are used. *Imerode()* function is used to erode the binary image and *imfill()* function is used to fill the noise. [1, 2] However, since there is still some noise on picture, image filtering method introduced in the class [1] is used by making the kernel for a box filter and filtering the image with it by using *imfilter()* function. After using the box filter, binary alpha map's values are not only 0 and 1, but since it is required to generate binary alpha map, the value smaller than 0.1 is converted to 0, and other values are converted to 1.

```
boxkern = ones(16);
boxkern = boxkern/sum(boxkern(:));
mask = imfilter(mask, boxkern);
for i = 1:a
        for j = 1:b
            if(mask(i,j)<0.1)
                mask(i,j) = 0;
            else
                mask(i,j) = 1;
            end
        end
    end
```

Code 2. Applying box filter, and generate the binary alpha map

The reference used for coding on MATLAB is cited on the last page "reference" from [1] to [6]. (Please refer to the more detail from the comments on the submitted code.)

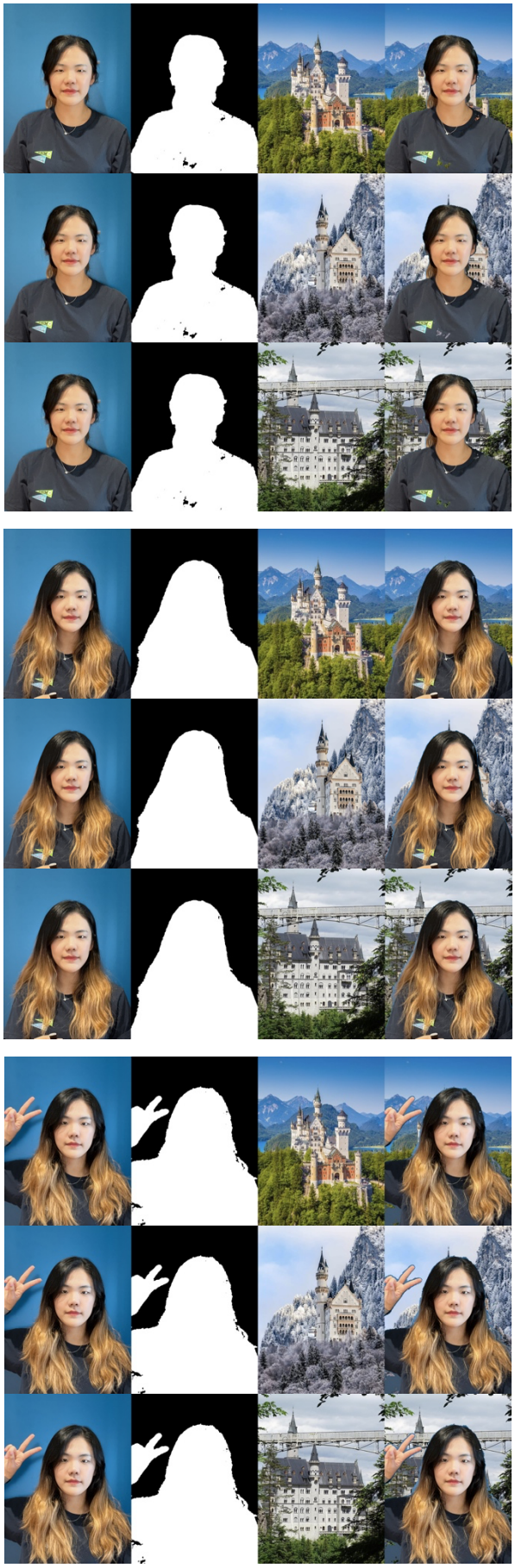


Figure 2. Computed image with binary alpha map. (input1, extracted alpha map, input2, output)

1.2 Improved algorithm:

The improved algorithm is for separating the foreground by non-binary alpha map. This is done by using the triangulation. Triangulation requires two images with the same foreground image, but with the different background colors. The alpha value for each pixel is calculated based on the formula (Formula 1) given in the paper "Blue Screen Matting" by Alvy Ray Smith.[7] Since this method computes alpha values based on the two background images and two images, by comparing each pixel's RGB values, it returns more accurate and reliable results.

$$\frac{(R_{f_1} - R_{f_2})(R_{k_1} - R_{k_2}) + (G_{f_1} - G_{f_2})(G_{k_1} - G_{k_2}) + (B_{f_1} - B_{f_2})(B_{k_1} - B_{k_2})}{(R_{k_1} - R_{k_2})^2 + (G_{k_1} - G_{k_2})^2 + (B_{k_1} - B_{k_2})^2}$$

Rf1 = Red value of Image 1	Rf2= Red value of Image 2	Rk1= Red value of background of Image 1	Rk2= Red value of background of Image 2
Gf1= Green value of Image 1	Gf2= Green value of Image 2	Gk1= Green value of background of Image 1	Gk2= Green value of background of Image 2
Bf1= Blue value of Image 1	Bf2= Blue value of Image 2	Bk1= Blue value of background of Image 1	Bk2= Blue value of background Image 2

Formula 1. Triangulation formula

Since this method requires two images with identical foreground and different background colors, two photos with blue and green background color are used to compute. However, due to the focus of camera, the color and the lightness of the image changed, and it causes the difference in colors of the background of original picture that I want to compute, and the color of background images that I took without any foreground image on it. Moreover, the foreground image can also have the similar RGB values with the background images, and on the other hand, since this generates non-binary alpha map, it caused transparency of foreground image, lowering its image resolution. To minimize this, the following Code 3 is used which emphasize the alpha value of pixel if it has the value that is larger than 0.8.

```
for p = 1:n
        for q = 1:m
            if(1-alpha(p,q)>0.8)
                alpha(p,q) = 0.6 * alpha(p,q);
            end
        end
    end
```

Code 3. Emphasizing some alpha values to improve computed image

The computed result is shown on Figure 3.

The reference used for coding on MATLAB is cited on the last page "reference" from [7] to [8]. (Please refer to the more detail from the comments on the submitted code.)

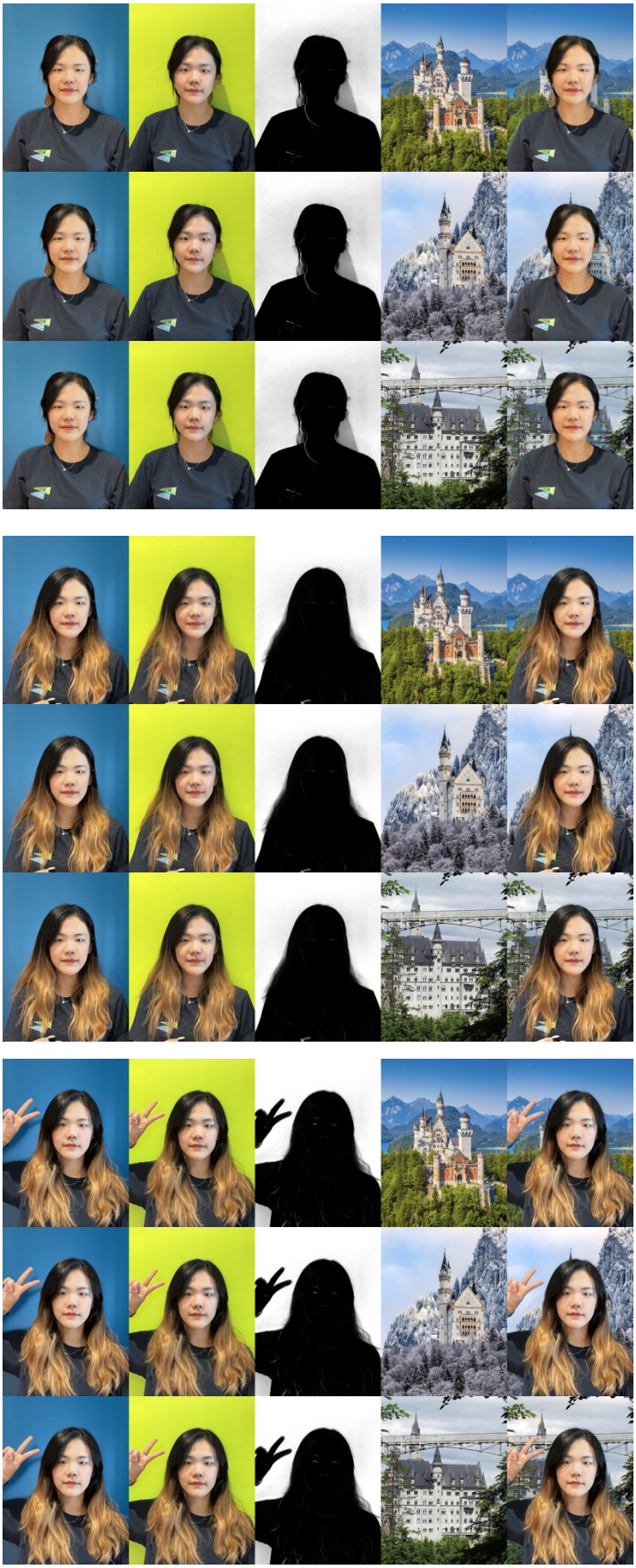


Figure 3. Result with non-binary alpha map. (input1, input1_2, extracted alpha map, input2, output)

1.3 Factors lowering the output quality:

- For both algorithm:

1. The difference of background colors between original images with foreground, and just background images without foreground.

: Both algorithms require the background images for each picture. However, not only the stability of camera isn't guaranteed, but also when there is human in front of the background, the camera is focused on the human, but when there is no human, the camera is focused on the background. This results in the different light/ brightness of images causing the different HSV/ RGB value.

: Due to the foreground figure (human, myself), the shadow occurs on the background. This caused the different HSV and RGB values between one that has a foreground and one that doesn't have it.

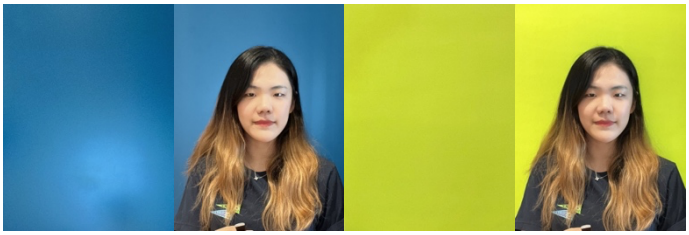


Figure 4. The difference of background colors for both blue (Left) and green (Right)

- For basic algorithm:

2. The messy hair

: Baby hair normally has low resolution and basic algorithm uses the hue values for each pixel. Therefore, for the basic algorithm, it is difficult to detect the messy part of the hair, causing the unclear and inaccurate outlines of foreground images.

Especially, because of this, the outline occurs on the computed images that make the edges between foreground and merged background images.

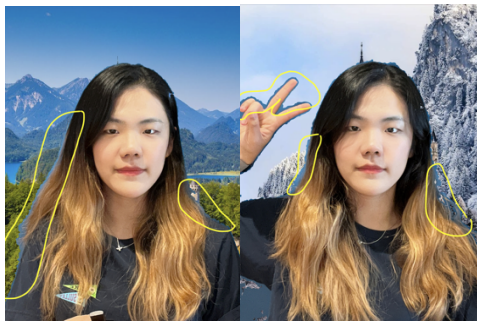


Figure 5. The distinguishable edges between foreground and background images (marked with yellow lines)

- For improved algorithm:

3. Small changes in factors such as position, posture, and light between same foreground images with the

different background colors can lower the quality.

: Two images with the same foreground, but different background colors (green and blue) were taken by changing the background colors manually with the green papers in front of the blue walls. No matter how fast we tried to take two photos, any delay can cause the slight difference between the foreground parts of these 2 images. This is because it is nearly impossible for me not moving even a little bit. On the other hand, since they have different background colors, the focus of the images can be changed automatically by camera, and this cause the difference in HSV and RGB values for each pixel. Moreover, the camera is not guaranteed to be at the same state.

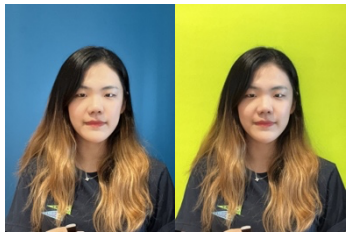


Figure 6. The difference of foreground figures of two images

1.3 Strength and weaknesses

1. Basic algorithm

1.1 Strength

1) Easy computation: compared to the improved algorithm, basic algorithm is much faster and easier to compute.

1.2 Weaknesses

1) The distinguishable edges between foreground and background images are occurred. This reduced the quality of computed images.

2) The messy outline of the foreground images causes the quality of outline, causing the sharp shape edges, rather than a smooth one.

2. Improved algorithm

1.1 Strength

1) Smooth outline, causing the smooth edges between combined foreground and background images.

: The alpha values of the edges vary, causing the smooth change from the foreground to the background.

2) The outline with the background colors (for my case, blue outline, shown on figure 5) isn't appeared with the improved algorithm.

1.2 Weaknesses

1) More computation is required compared to the basic algorithm.

2) Two images with same foreground, but different background colors are required. This is not common in the real life, so it is usually difficult to adopt this algorithm.

Part 2: Image Blending

2.0 Prepare Data

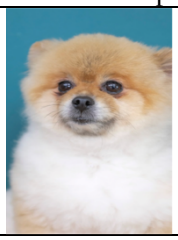
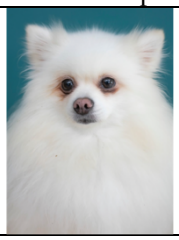
		
Level 1. Yellow Maple leaf	Level 1. Red Maple leaf	Mask
		
Level 2. Dog1	Level 1. Dog2	
		
Level 3. Myself	Level 3. Singer (Irene)	

Figure 7. The prepared data for part 2

* The used background images were downloaded from [12] to [15].

2.1 General Explanation of Algorithm

By constructing Laplacian pyramid of two images and merging with the Gaussian pyramid's image for the mask, the perfectly blended images of two different foreground objects can be constructed. As explained in the class, two images can be merged without any visible distinction or edge between them by smoothly blending the low-frequency color and rapidly blending for the higher-frequency. [16]

The levels of pyramid is determined by following code:

```
levels_1 = floor(log2(a/16));
levels_2 = floor(log2(b/16));
levels = min(levels_1, levels_2);
```

This is because, it was difficult to determine the levels manually, and want to make it more flexible. Thus, by calculating the integer value for how many each width and heigh can be divided by 2, the levels are calculated. On top of it, if the size of the image is in decimal numbers, there can be the loss of image information,

so the image is resized to ensure that it can be divided by 2 for the number of levels, and for each step, the width and heigh are in integer values.

For the kernel, binomial kernel is used, rather than a real Gaussian kernel as given in the class. [16]

$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Figure 7. The used kernel [16]

*Overall ideas about image pyramid and blended image are obtained by the given lecture note [16] and by the explanation on [17].

2.2 Obtained results

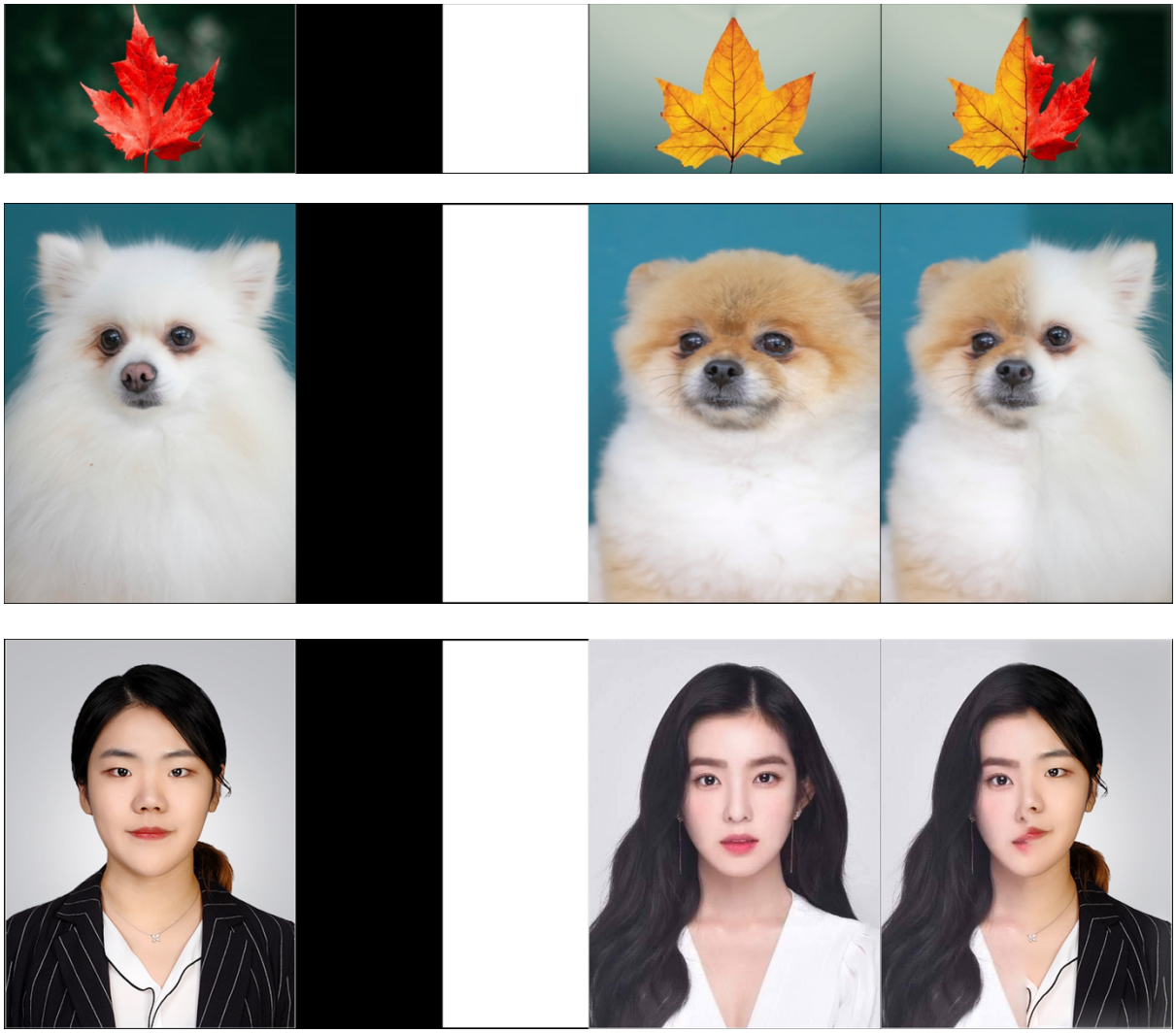


Figure 8. The obtained data (input1, kernel, input2, output)

2.3 Identifying the factors that affect image quality

There can be many factors that can affect image qualities.

1. High color variation between two images

When there is high color variation between two images, the result can be unrealistic, with the mixed blurred colors of two images. For example, as the following Figure 9 shows, for the middle part, there is high color difference between my hair color (yellow) with the t-shrit (navy) and it causes the blurred yellow color on the middle. However, since it is unrealistic, this can be the factor which can lower the image quality.

Therefore, I intentionally chose the images that have similar background colors, and not vivid feature for the background, so their background can be merged smoothly as well. Moreover, images were chosen to have less color variation for the middle part when I merge them together.

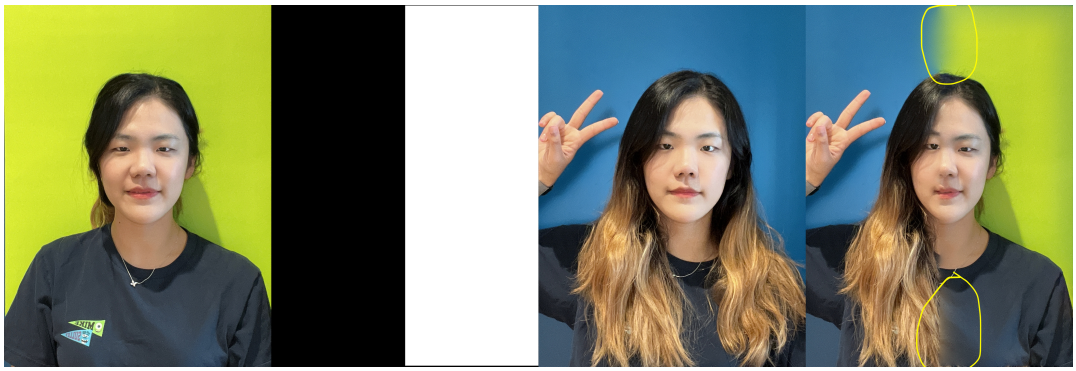


Figure 9. The obtained data for showing the high color difference between two input images

2. The quality of mask

To get a natural and realistic result of merged, blended image, we must pick the appropriate mask. For example, as the Figure 9 shows, as the mask was not on the center of my face, the merged image is not realistic nor shows the appropriate figure. The mask should be chosen reasonably considering all the features on the images.

Therefore, by considering the chosen images, I chose the mask that can work perfectly and returns the high quality.

3. The resolution and quality of image

To get a natural and realistic result of merged, blended image, we must pick the images of high enough resolution / quality. For example, as the Figure 10 shows, input 1 image has high quality, but input 2 has very low image quality. With this fact, after blending two images, the quality of computed image is very low.

Therefore, all the chosen images have high enough resolution and quality, so they can return the high quality merged images.

Figure 10. The obtained data for showing the high color difference between two input images

By considering these factors, I intentionally chose the image sets that has good resolution and quality, smooth color variance in the middle, similar features. Moreover, I carefully chose the mask that shows the high quality computed images by considering the chosen images.

2.4 Extra computation

To combine part 1 and part 2, with the binary alpha map, I could have computed the following results.

With this computation, the obtained results show higher quality, especially for the maple leaf and portrait one.

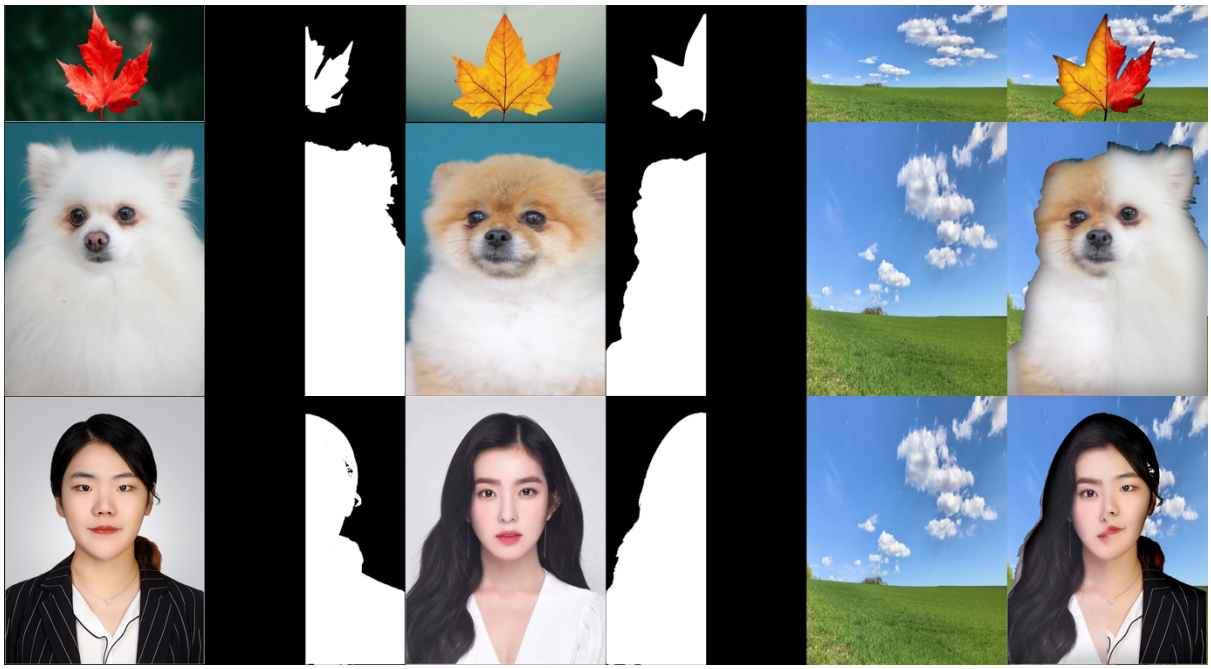


Figure 11. Merged blended images with the background change

Reference

[1] Yin, K. (n.d.). *Cmpt 361 Matlab Lab 1*. Lecture.

[2] Yin, K. (n.d.). *CMPT 361 MATLAB Tutorial*. Lecture.

[3] *I want to display image from cell array* . -. (n.d.). Www.mathworks.com. Retrieved June 5, 2022, from https://www.mathworks.com/matlabcentral/answers/236725-i-want-to-display-image-from-cell-array

[4] *How can I split an hsv image into separate h,s,v components?* -. (n.d.). Www.mathworks.com. https://www.mathworks.com/matlabcentral/answers/248090-how-can-i-split-an-hsv-image-into-separate-h-s-v-components

[5] 비스카이비전. (2017, October 25). *[색공간] HSV 색 공간을 활용해서 특정 색깔의 물체만 검출하기 (matlab 소스코드 포함)*. Bskyvision. https://bskyvision.com/46

[6] *Morphological structuring element - MATLAB*. (n.d.). Www.mathworks.com. https://www.mathworks.com/help/images/ref/strel.html

[7] Smith, A. R., & Blinn, J. F. (1996). Blue screen matting. Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '96. https://doi.org/10.1145/237170.237263

[8] *how to save/write images using for loop?* -. (n.d.). Www.mathworks.com. Retrieved June 5, 2022, from https://www.mathworks.com/matlabcentral/answers/115539-how-to-save-write-images-using-for-loop

[9] *Neuschwanstein Castle covered in winter snow. - Picture of Hohenschwangau, Swabia - Tripadvisor*. (n.d.). Www.tripadvisor.in. Retrieved June 5, 2022, from https://www.tripadvisor.in/LocationPhotoDirectLink-g187307-i302040284-Hohenschwangau_Swabia_Bavaria.html

[10] Redirect Notice. (2022). Google.ca. https://www.google.ca/url?sa=i&url=https%3A%2F%2Fwww.mikesbiketours.com%2Fmunich%2Four-tours%2Fcastle-tours.html&psig=AOvVaw30AHLT6wy0ykmu2djT_7YU&ust=1654548052108000&source=images&cd=vfe&ved=0CAwQjRxqFwoTCOC_hvmVl_gCFQAAAAAdAAAAABAD

[11] Neuschwanstein Castle. (n.d.). Schwangau.de. https://en.schwangau.de/sightseeing/neuschwanstein-castle/

[12] *아이린 증명사진*. (2022, June 15). 인스티즈(Instiz). https://www.instiz.net/pt/6712032

[13] *Maple leaf Images, Stock Photos & Vectors*. (n.d.). Shutterstock. Retrieved June 14, 2022, from https://www.shutterstock.com/search/maple+leaf

[14]src='https://secure.gravatar.com/avatar/e4230f9364587bb11ac8f3a40c196b95?s=96, S. N. I. H. alt='', #038;d=mm,

Srcset=’https://Secure.gravatar.com/Avatar/E4230f9364587bb11ac8f3a40c196b95?s=192, 038;r=g’, #038;d=mm, Paulo, 038;r=g 2x’ class=’avatar avatar-96 photo’ height=’96’ width=’96’ itemprop="image" /> A. N. I. H. é referência no setor de hotelaria! M. presença em importantes capitais como S., Janeiro, R. de, Horizonte, B., Curitiba, Salvador, Caldas, P. A. e R. A. empresa é 100% brasileira e a maior de capital nacional do país F. em 1970 em P. de, Unidades, A. C. C. M. D. 50, & Estados!, E. 23 D. E. S. (2022, June 3). *Ainda dá tempo de planejar uma viagem de outono*. Blog Hotéis Nacional Inn. https://www.blog.nacionalinn.com.br/ainda-da-tempo-de-planejar-uma-viagem-de-outono/

[15] *곤지암 애견운동장 시티독 (구 천호 시티독 애견카페)*. (n.d.). 네이버 블로그 : 수블로그. Retrieved June 14, 2022, from https://m.blog.naver.com/PostView.nhn?blogId=sublogo0o&logNo=222118337933&categoryNo=31&proxyReferer=

[16] Yin, K. (n.d.). *Cmpt 361 Image Pyramid*. Lecture.

[17] Zhao, M. (2020, May 14). *Image Blending Using Laplacian Pyramids*. Medium. https://becominghuman.ai/image-blending-using-laplacian-pyramids-2f8e9982077f