Assignment 3

Mu-Ruei Tseng

February 27, 2024

1 Overview

In this assignment, we focused on two major tasks: generating hybrid images using Gaussian and Laplacian filters and implementing pyramid blending to merge two images more smoothly.

2 Task 1

2.1 Method

To generate the hybrid image, we prepare two images, denoted as I_1 and I_2. We begin by defining two Gaussian kernels, G_1 and G_2, with standard deviations σ_1 and σ_2, respectively. It is recommended to determine the size of each kernel using the formula $k = 6\sigma - 1$, where σ is the corresponding standard deviation of the kernel.

Next, we generate the 2D Gaussian matrix by taking the outer product of a 1D Gaussian vector. The 1D Gaussian vector for a given σ is defined by the equation:

$$x_i(\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \tag{1}$$

Here, $\vec{x}$ is a vector with elements $[1, 2, \ldots, k]$, and μ represents the mean of $\vec{x}$. Consequently, the 2D Gaussian matrix $G(\sigma)$ is given by:

$$G(\sigma) = \vec{x}^{\top} \otimes \vec{x} \tag{2}$$

where $\otimes$ denotes the outer product, and $\vec{x}^{\top}$ is the transpose of $\vec{x}$.

This matrix can then be used to apply a Gaussian blur to I_1 and I_2, after which the images are combined to produce the hybrid image.

Now we have the Gaussian filter, we can use it to create the Laplacian filter $L(\sigma)$ by:

$$L(\sigma) = I_m - G(\sigma) \tag{3}$$

where I_m is the impulse filter.

After having both the Gaussian and Laplacian filters, we can therefore generate our hybrid image:

$$I_{hybrid} = G(\sigma_1) * I_1 + L(\sigma_2) * I_2 \tag{4}$$

After generating the hybrid image, I also normalize it so that it can be within the range of 0 and 1.

$$I_{hybrid} = \frac{I_{hybrid} - \min(I_{hybrid})}{\max(I_{hybrid}) - \min(I_{hybrid})} \tag{5}$$

2.2 Result

Here we first show the sample hybrid image result using Monroe(I_1) and Einstein(I_2), see Figure 1.

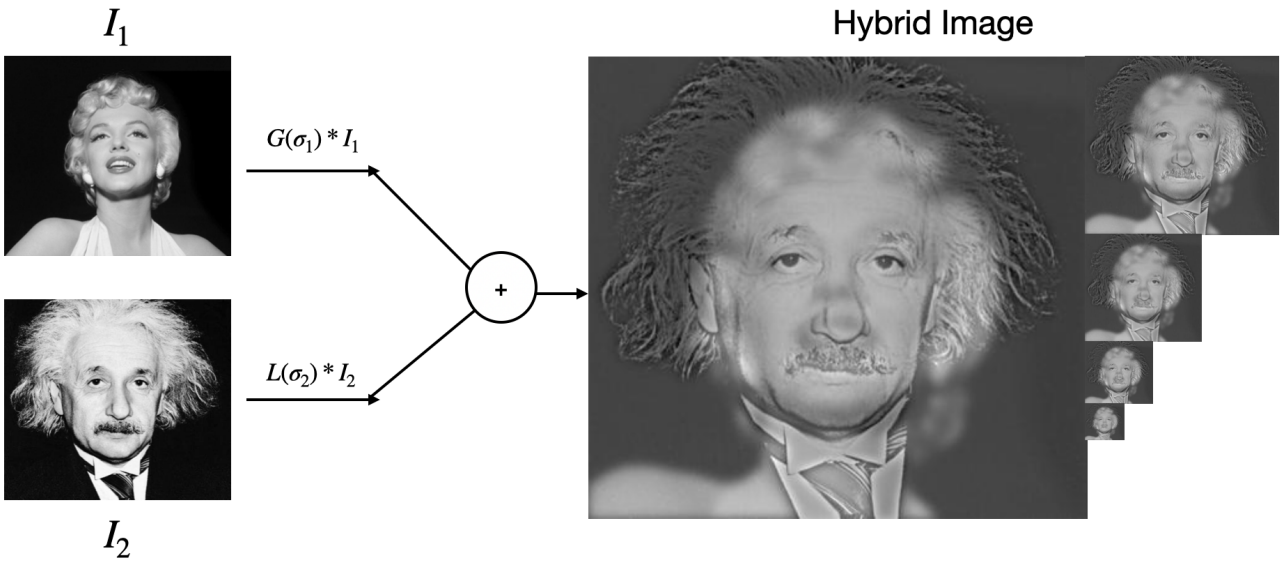

Figure 1: Hybrid image result using Monroe(I_1) and Einstein(I_2). We set both σ_1 and σ_2 to 4 and the corresponding kernel size is 23×23. The two images are aligned using the eyes' position.

I have also experimented with the hybrid image on different input samples. Here are the results and their corresponding configurations. See Figure 2 and 3

From the results, we can observe that as the size of the hybrid image increases, it more closely resembles I_2, which is the image processed with the Laplacian (High-Pass) filter. Conversely, as the hybrid image decreases in size, it begins to look more similar to I_1, the image that has been subjected to the Gaussian (Low-Pass) filter. However, fine-tuning the standard deviation is necessary to achieve the most desirable result. Here, we also show an ablation study on experimenting with different σ_1 and σ_2. See Figure 4. Given the same σ_1, increasing σ_2 emphasizes the high-frequency areas more, making the larger image bear a greater resemblance to I_2. Conversely, using a consistent σ_2 while increasing σ_1 reduces the high-frequency components, resulting in a smoother overall image. Therefore, finding a balance between σ_1 and σ_2 is vital in generating better hybrid images.

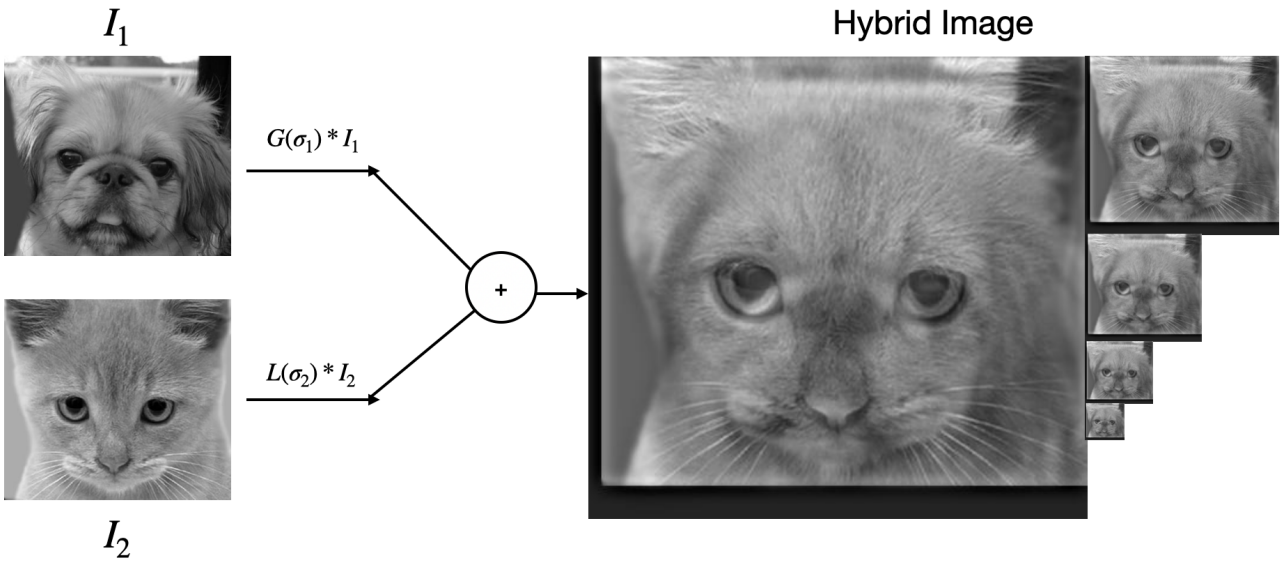

Figure 2: Hybrid image result using dog(I_1) and cat(I_2). We set σ_1 as 3 and σ_2 as 7. The kernel sizes are 8×8 and 41×41 respectively. The two images are the same size; therefore, no alignment is performed.

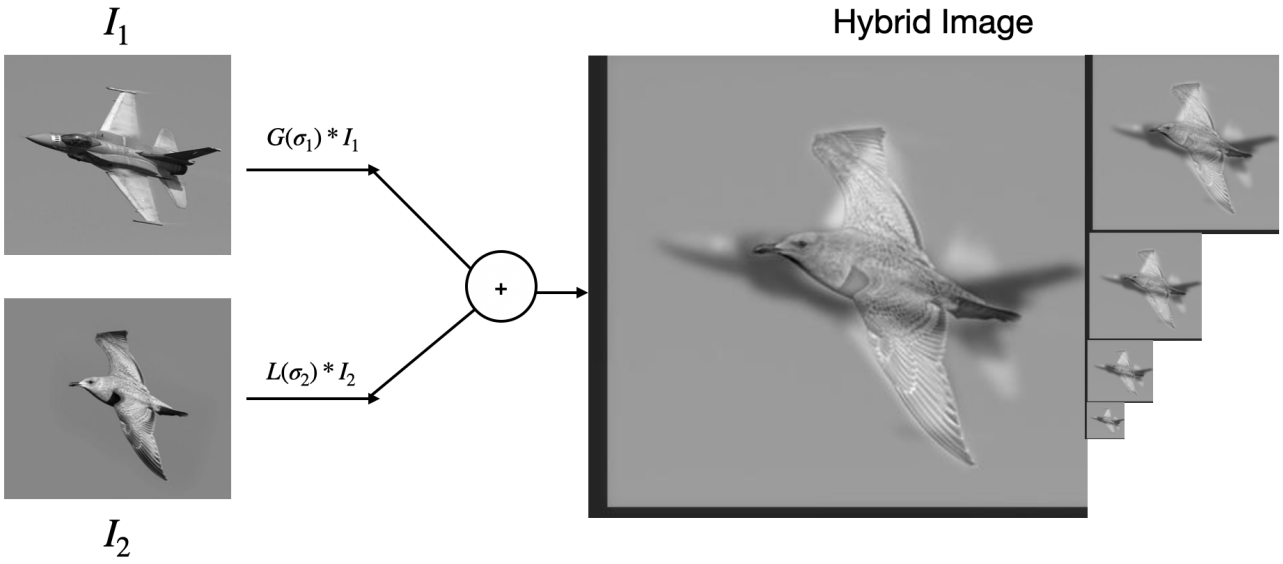

Figure 3: Hybrid image result using plane(I_1) and bird(I_2). We set both σ_1 and σ_2 to 4 and the corresponding kernel size is 23×23. The two images are the same size; therefore, no alignment is performed.

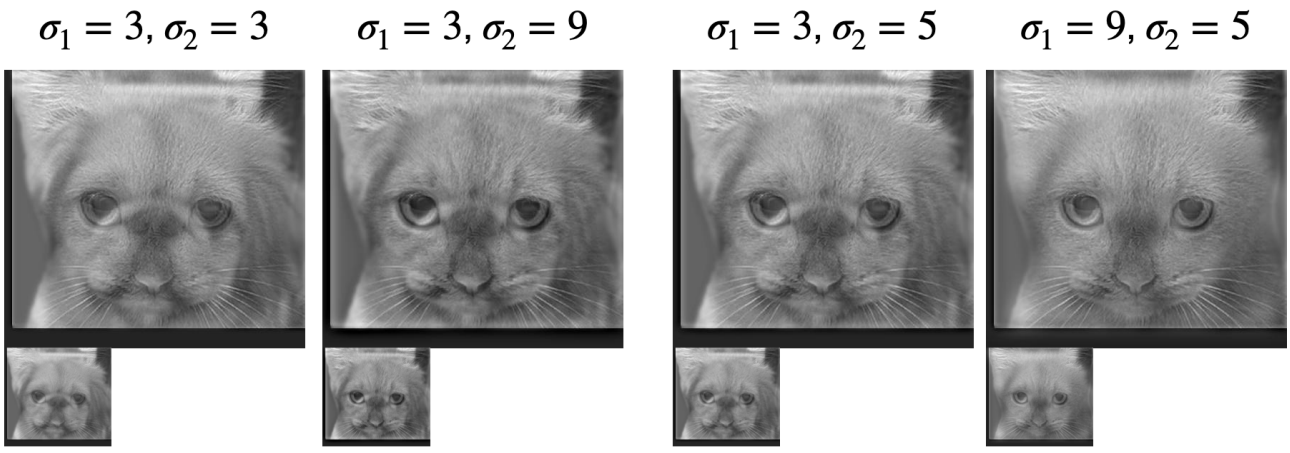

Figure 4: Experimenting on different sets of σ_1 and σ_2

3 Extra Credit

For the extra credit, we attempt to implement image blending with color images. To adapt our previous algorithm for color images, we apply the Gaussian and Laplacian filters to each color channel. We experiment with four different settings: combining RGB I_1 with RGB I_2, Gray I_1 with RGB I_2, RGB I_1 with Gray I_2, and Gray I_1 with Gray I_2. Here we use the dog (I_1) and cat (I_2) hybrid image as samples with $\sigma_1 = 3$ and $\sigma_2 = 7$. See Figures 5, 6, 7, 8.

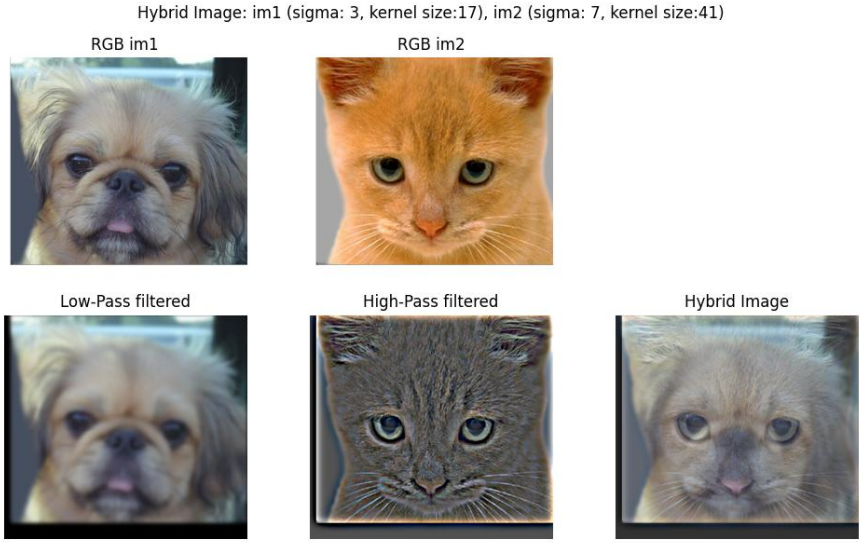

Figure 5: The result on using RGB I_1 and RGB I_2

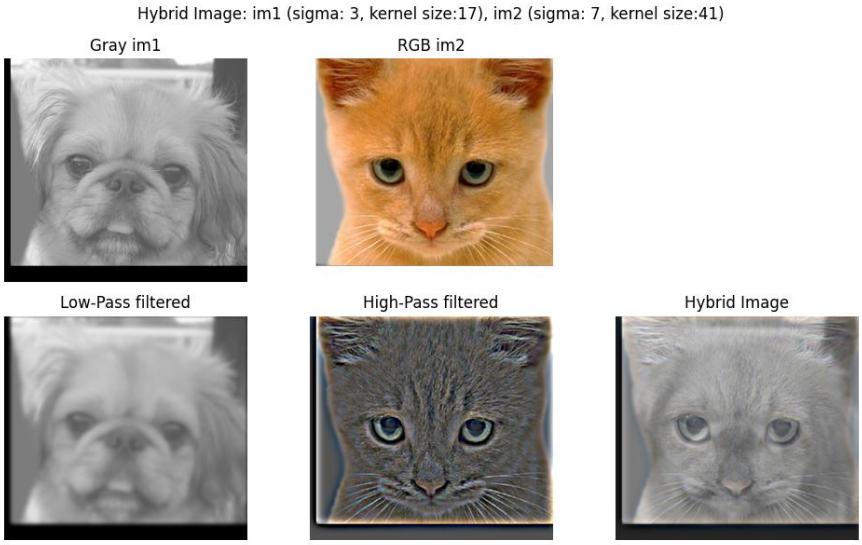

Figure 6: The result on using Gray I_1 and RGB I_2

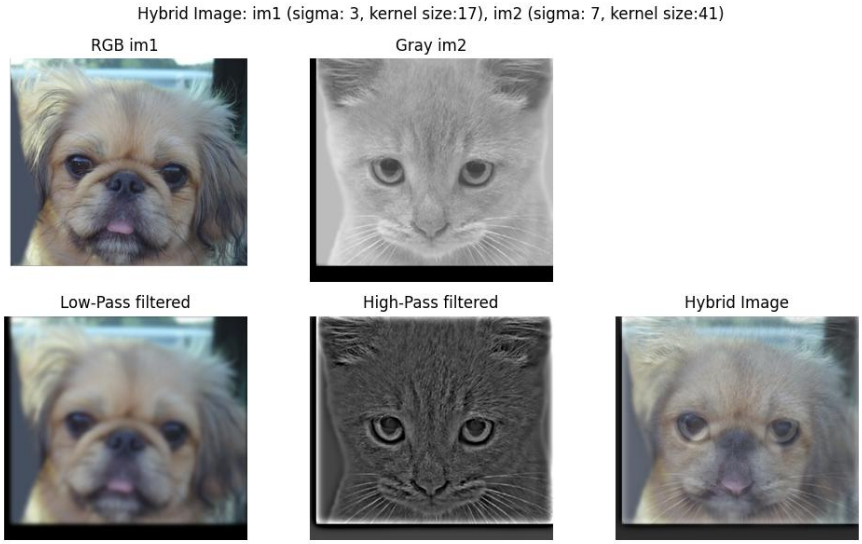

Figure 7: The result on using RGB I_1 and Gray I_2

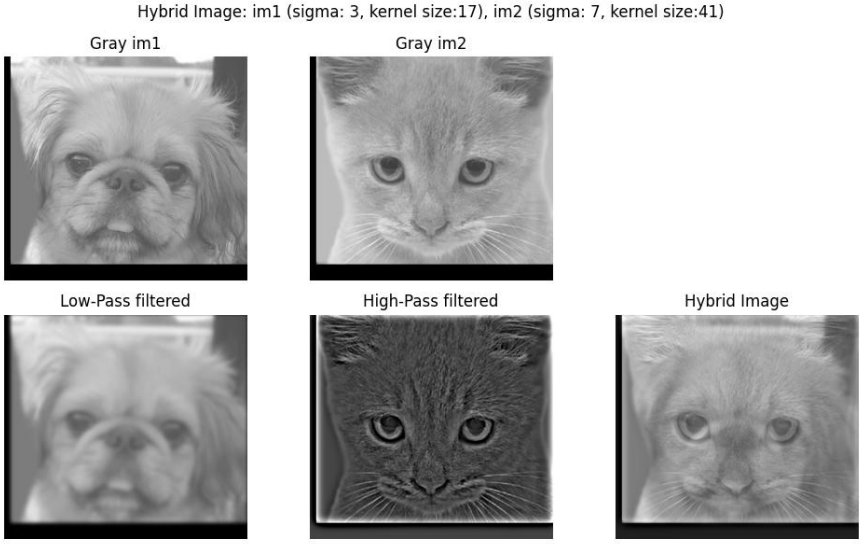

Figure 8: The result on using Gray I_1 and Gray I_2

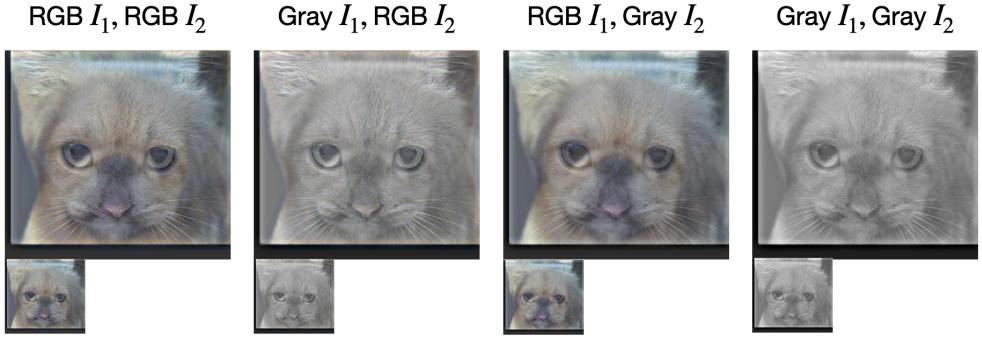

Figure 9: The result comparison under the four different settings.

From Figure 9, we observe the comparative results of using grayscale versus RGB images to generate the hybrid image. It is evident that when RGB is used for I_2, the resulting hybrid image tends to retain more of the contextual information from I_2 compared to when a grayscale version of I_2 is utilized. This phenomenon likely comes from the fact that color images contain more high-frequency information compared to their grayscale counterparts. Therefore, for the creation of a high-quality hybrid image, selecting Grayscale I_1 with RGB I_2 is advisable.

4 Task 2

In this section, we will implement the pyramid blending function to merge two images more smoothly. Pyramid blending involves four main parts: creating the Gaussian pyramid from the mask, creating the Laplacian pyramid from the source and target images, merging the Gaussian and Laplacian pyramid, and collapsing the merged pyramid to generate the final output image.

- Gaussian Pyramid is generated by applying the Gaussian filter and then performing subsampling.

$$G[l+1] = \text{subsample}(g * G[l]) \tag{6}$$

  where $G[l+1]$ represents the l-th level of the Gaussian Pyramid, g is the Gaussian filter, and subsample$(\cdot)$ is the operation of subsampling by taking a stride of 2 from the image at $G_M[l]$.

- Laplacian Pyramid is generated by setting the last layer as the same as the last layer of the Gaussian Pyramid, and then for each level, it calculates the difference between the current Gaussian level and the expanded version of its subsequent Gaussian level. This process captures the image details lost between levels in the Gaussian Pyramid.

$$L[l] = G[l] - \text{expand}(G[l+1]) \tag{7}$$

  where $L[l]$ represents the l-th level of the Laplacian Pyramid, and expand$(\cdot)$ is the operation of upsampling followed by a Gaussian filter to interpolate the missing values. The last layer of the Laplacian Pyramid is set equal to the last layer of the Gaussian Pyramid layer as there are no higher resolution levels to subtract from.

- Combine the source and target Laplacian pyramids:

$$L_C[l] = G_M[l] \times L_S[l] + (1 - G_M[l]) \times L_T[l] \tag{8}$$

  where L_S and L_T are the Laplacian pyramid from the source and target images and G_M is the Gaussian pyramid from the mask.

- Collapsing the merged pyramid: Collapsing the merged pyramid is accomplished by iteratively resizing the image from the lower level to twice its size and then adding it to the current level

After collapsing the merged pyramid, I clip the output image to be within the range of 0 and 1.

4.1 Result

Here we show the pyramid blending using the technique we mentioned above.

Sample output The blending result of using a half-white, half-black mask to merge the apple image with the orange image is shown in Figure 10. We can observe that the blend barely has a distinct edge after pyramid blending.

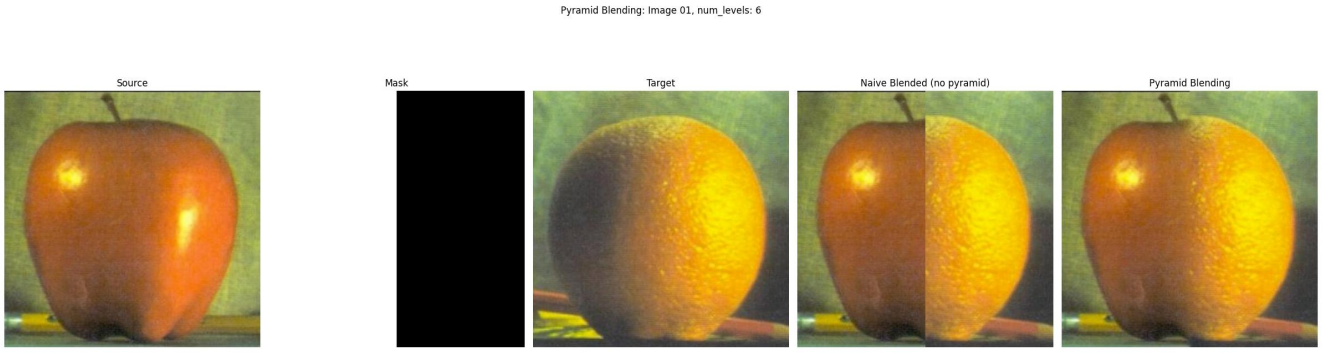

Figure 10: The pyramid blending result on sample images (orange and apple). The number of levels $k = 6$

Nice blended image In this part, we blend a person's eye onto the other person's hand. I masked out the eye part using the given GetMask function and adjusted the position of the eye in the source image so that it could fit perfectly on the hand. See Figure 11.

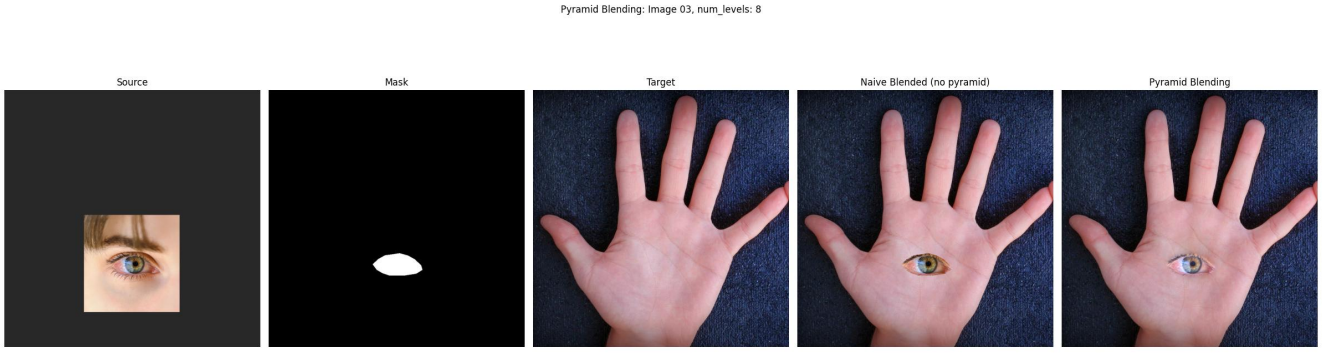

Figure 11: The pyramid blending results in merging an eye with a hand. The number of levels $k = 8$

Failure Case In this section, we attempted to blend an actor's face with the Mona Lisa. We masked out the face region in the source image and aligned it with the face position in the Mona Lisa. However, the result was not as satisfactory as in previous cases due to the substantial

difference in color styles between the source and target images. Although pyramid blending enables smoother blending of two images at their borders, its effectiveness can be significantly reduced when the colors of the two images are markedly different.

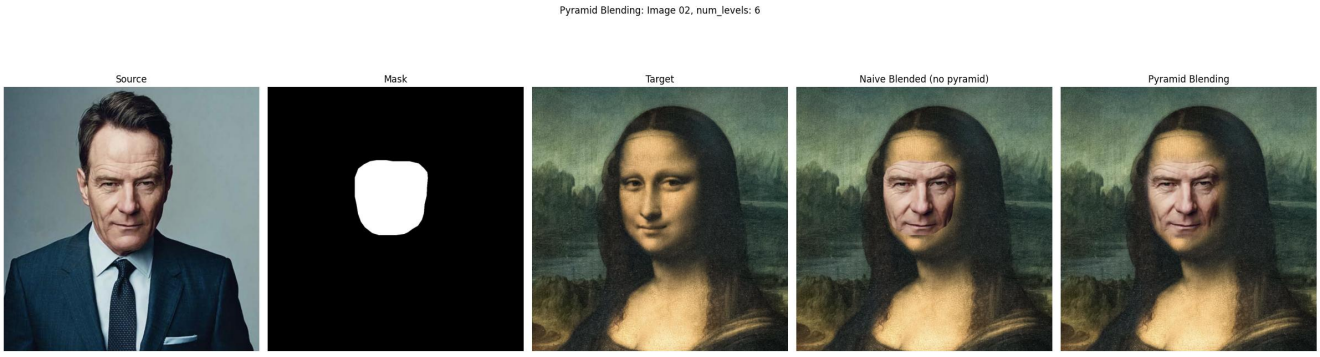

Figure 12: The pyramid blending results in merging the actor with a Mona Lisa. The number of levels $k = 6$