

DIGITAL IMAGE PROCESSING ASSIGNMENT 3

Muhammed Görkem KOLA - 2200765032, Hacettepe University

11/12/2022

What is Expected?

In this assignment we are asked to blend images using gaussian and laplacian pyramids.

What I did

I picked 11 images (5 for background, 6 for foreground) and blended them.

Listing 1: Gaussian Function

```
1 # gaussian pyramid function -> takes an image and a number that shows how ←  
    many level will be deep gaussian pyramid, as arguments and returns a ←  
    gaussian pyramid.  
2 def gaussian(img, number):  
3     imgs = [img]  
4     for i in range(number):  
5         img = cv.pyrDown(img)  
6         imgs.append(img)  
7     return imgs
```

Listing 2: Laplacian Function

```
1 # laplacian pyramid function -> takes a gaussian image pyramid and a ←  
    number that shows how many level will be deep laplacian pyramid, as ←  
    arguemnts and returns a laplacian pyramid.  
2 def laplacian(gauss_imgs, number):  
3     imgs = [gauss_imgs[-1]]  
4     for i in range(number - 1, -1, -1):  
5         height, width, d = imgs[i + 1].shape  
6         size = width, height  
7         expanded_gauss = cv.pyrUp(gauss_imgs[i + 1], dstsize=size)  
8         laplace = cv.subtract(gauss_imgs[i], expanded_gauss)  
9         imgs.append(laplace)  
10    return imgs
```

Listing 3: Masking Function

```
1 # dynamic masking function using otsu method -> It takes and image as an ←
  argument and returns a masked image. (It has a disadvantage when image←
  's main character is darker then background.) ←
2 def mask(img):
3     gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
4     T, threshold = cv.threshold(gray, np.median(gray), 255, cv.←
      THRESH_BINARY_INV | cv.THRESH_OTSU)
5     thresh_with_blur = cv.medianBlur(threshold, 15, 0)
6     return thresh_with_blur
```

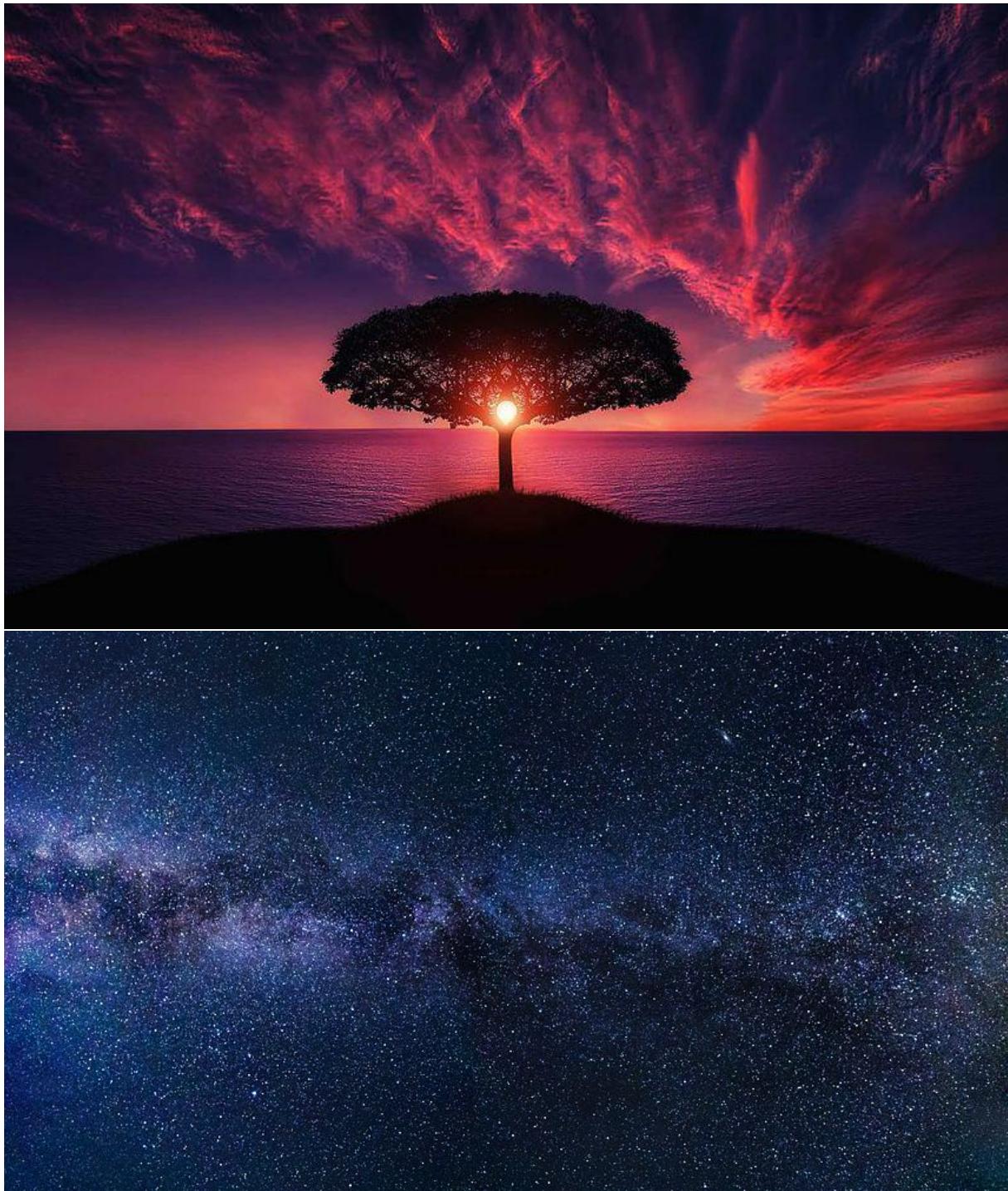
Listing 4: Masking Function

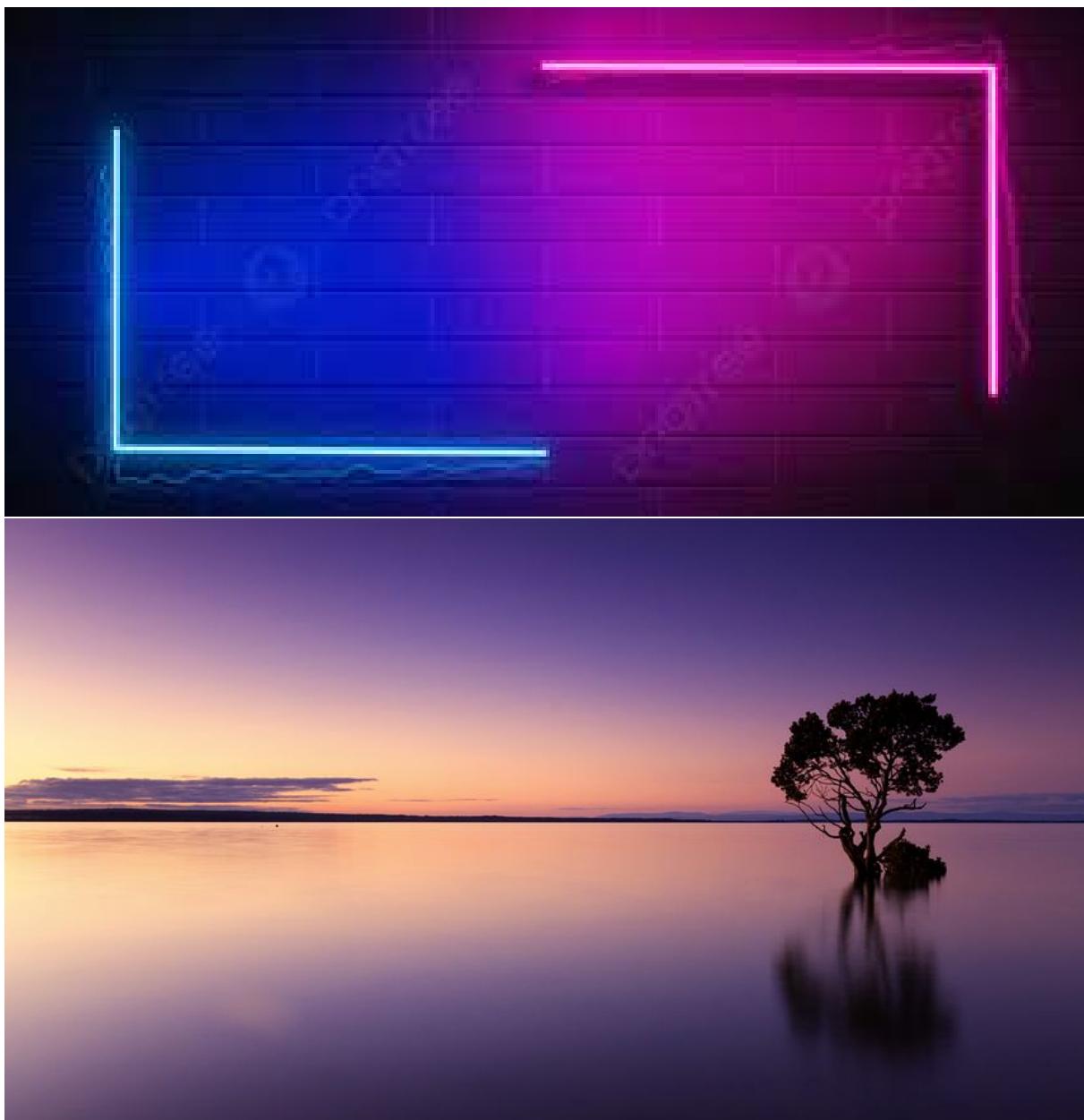
```
1 # blending function -> takes 2 laplacian image pyramid and a gaussian mask←
  pyramid as arguments and returns a blended image pyramid. ←
2 def blend(imgs1, imgs2, masks):
3     blendeds = []
4     for (la, lb, lmsk) in zip(imgs1, imgs2, masks):
5         ratio = np.count_nonzero(lmsk) / (lmsk.shape[0] * lmsk.shape[1])
6         # trying to solve choosing background
7         if np.mean((1 - lmsk) * lb) / (1 - ratio) < np.mean(lmsk * lb) / ←
            ratio:
8             blended = (1 - lmsk) * lb + lmsk * la
9         else:
10            blended = lmsk * lb + (1 - lmsk) * la
11        blendeds.append(blended)
12    return blendeds
```

Listing 5: Masking Function

```
1 # reconstructing function -> It takes blended image pyramid as argument ←
  and returna reconstructed pyramid. ←
2 def reconstruct(imgs):
3     top = imgs[0]
4     lst = [top]
5     number = len(imgs) - 1
6     for i in range(number):
7         height, width, d = imgs[i + 1].shape
8         size = width, height
9         lap_expanded = cv.pyrUp(top, dstsize=size)
10        top = cv.add(imgs[i + 1], lap_expanded)
11        lst.append(top)
12    return lst
```

Images Used for Background







Images Used for Foreground









How different levels affect results

[t]



Figure 1: Level 1

This image is not blended good. The astronaut seems as a different object.

Analyze of It

The higher pyramid level is good for blending images, but if you want to make foreground image prominent you can choose a lower pyramid level.

Results

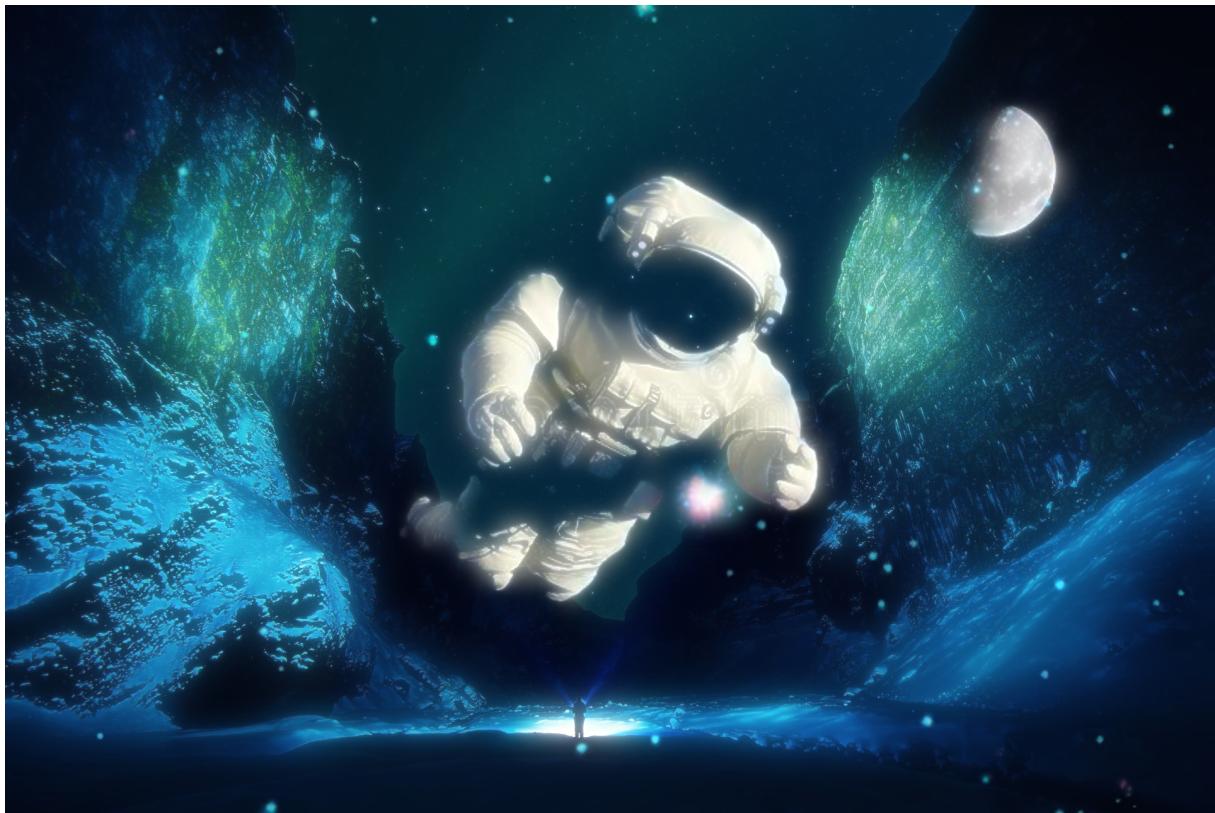


Figure 2: Level 5

This image is blended good. The astronaut seems a part of image.

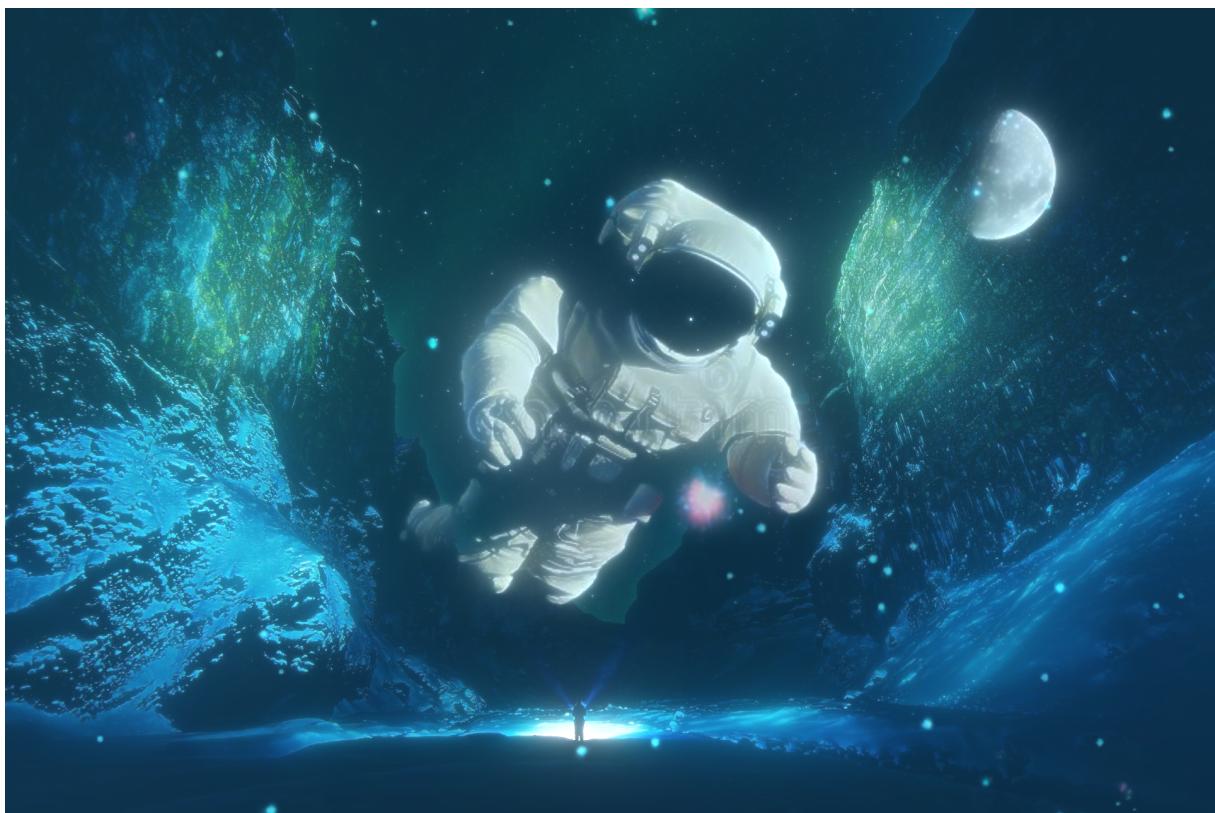


Figure 3: Level 10

This image is blended so good. The astronaut seems a part of image.



Figure 4: Results 1

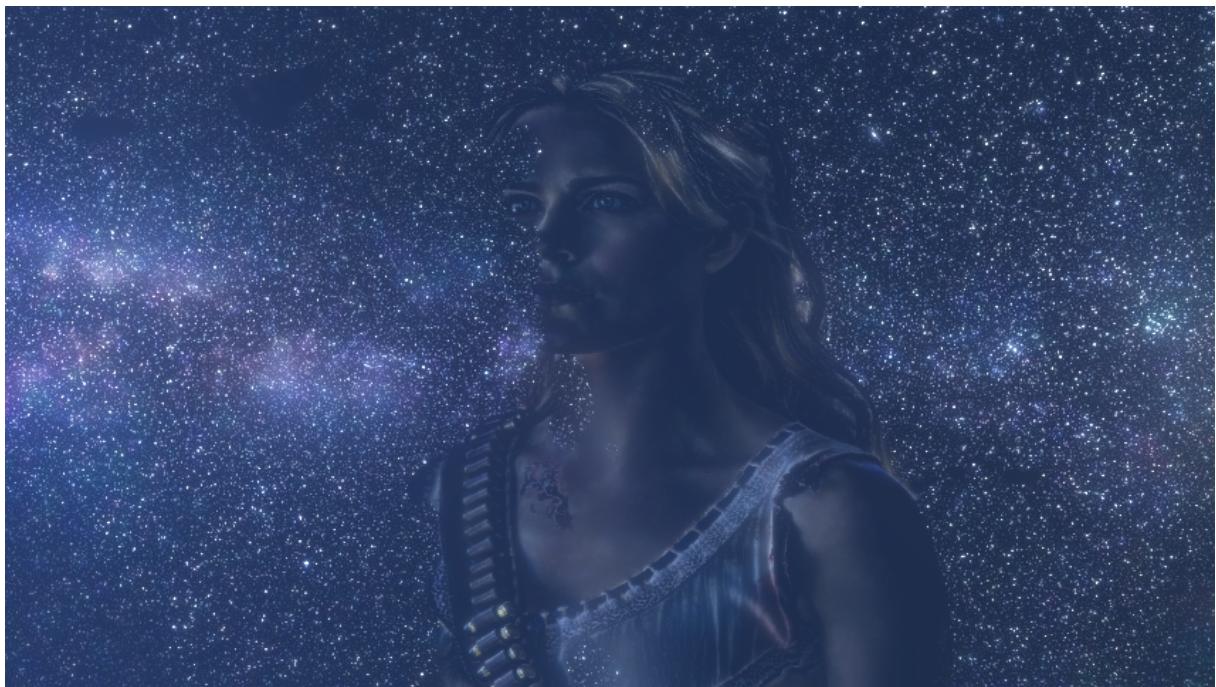


Figure 5: Results 2

Dolores's skin color is changed because of choosing higher pyramid level.



Figure 6: Results 3

Itatchi is disappeared because mask function chose the background and my



Figure 7: Results 4

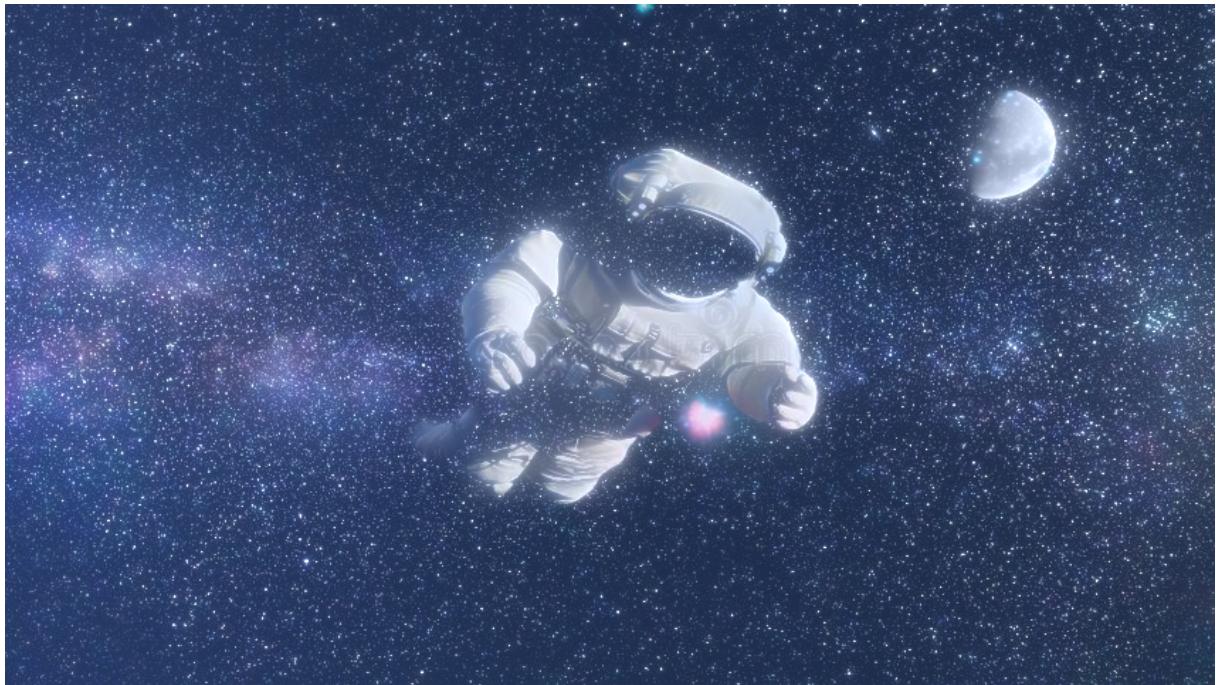


Figure 8: Results 5

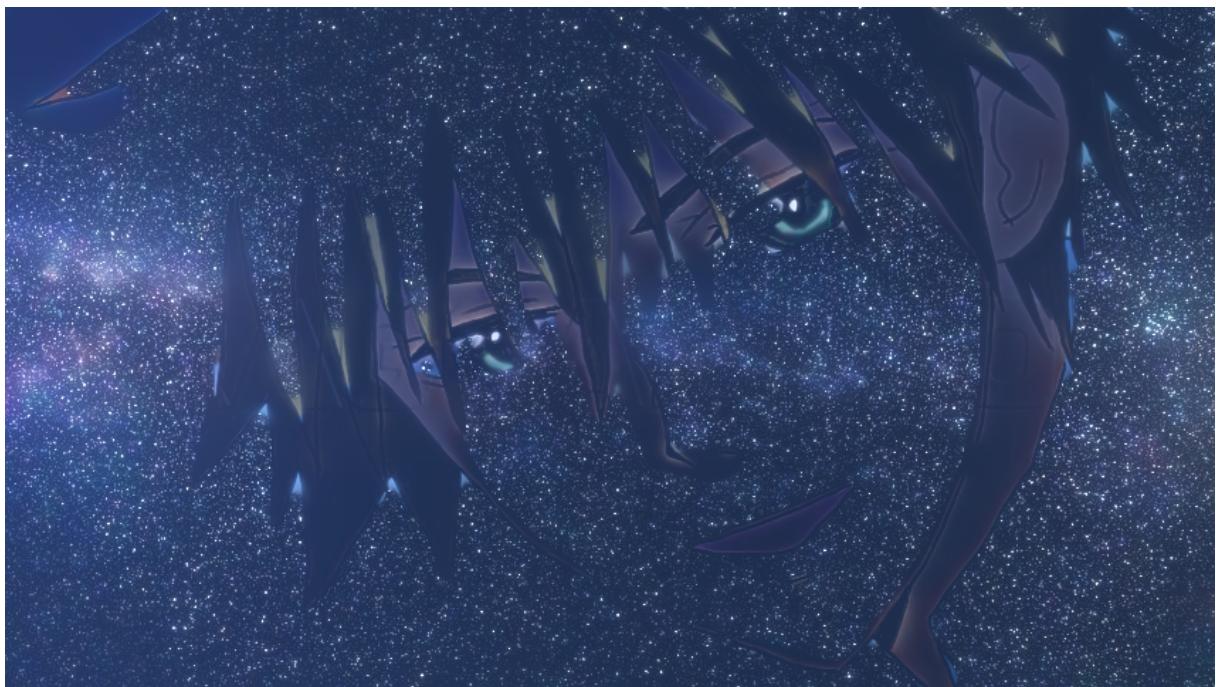


Figure 9: Results 6

As a result of my choice in blending function the face is disappeared.

Analyze of Results

Some of results are good but some of them are bad either because of my code or background brightness. But though they seem good blended.