

A World Travel

By Chromakey



20211060 오서현

20201043 김선경

20190365 김찬주

Github : https://github.com/hyunneee/vmp_travel

1. Experiment

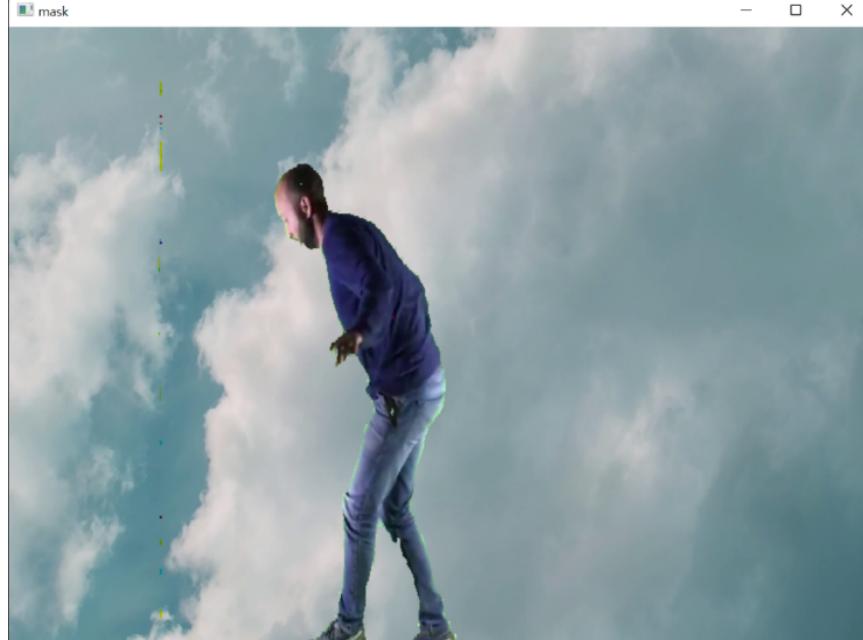
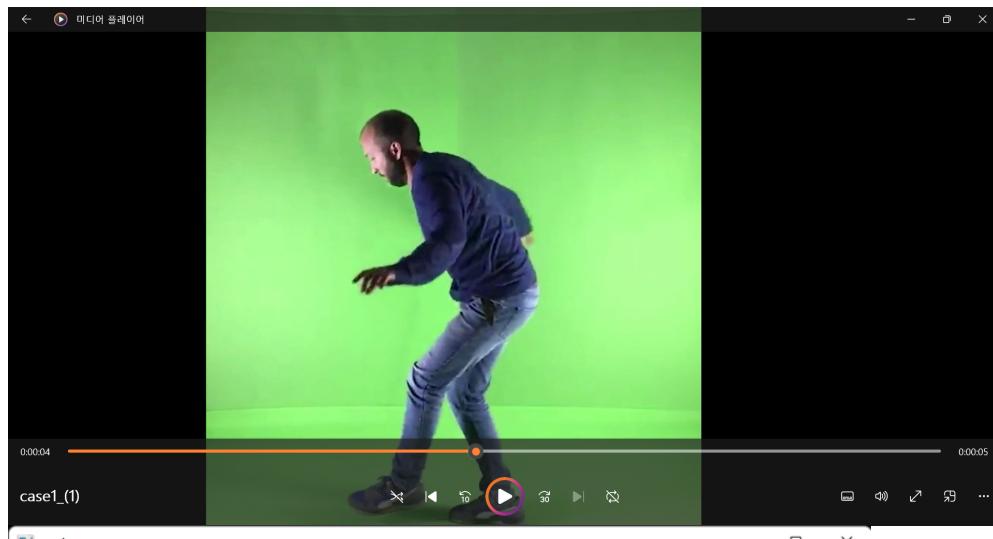
To verify that our code runs well, we tried to experiment with the code through various input images. Images taken from a green background that can be found through research can be divided into three categories.

- <1> Source taken in real life
- <2> Source whose existing background was removed and green background was applied.
- <3> Various effects sources with a green background Image taken in where the background scene is of a unique color

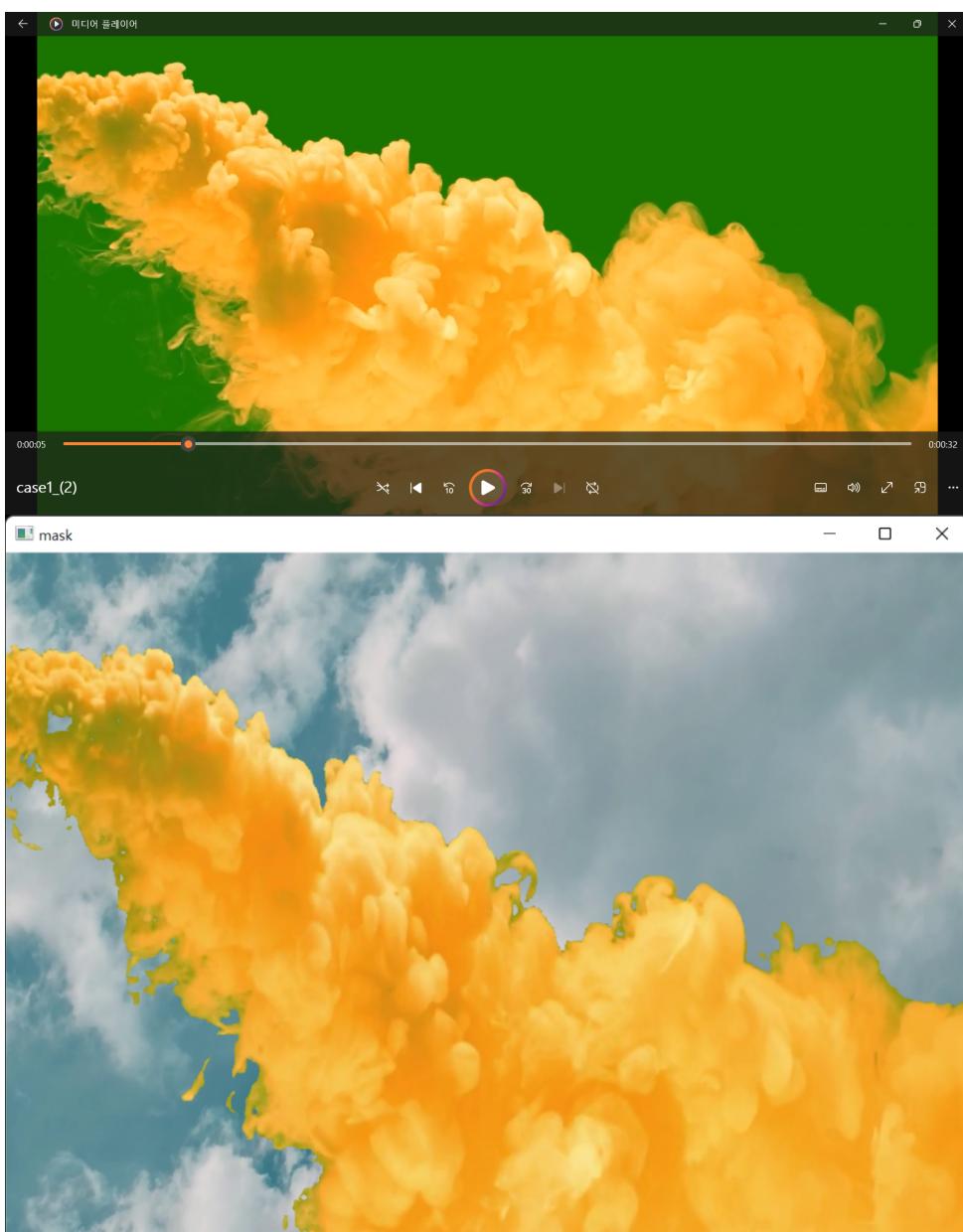
Therefore, we tried to experiment with our code by selecting two images for each category.

<1> Source taken in real life

- Case1_(1)



- Case1_(2)



It can be confirmed that both images are detected clearly without additional code modification.

<2> Source whose existing background was removed and green/blue background was applied.

- Case2_(1) Apply Green



Result



- Case2_(2) Apply Blue



```
# define range of blue color in HSV
lower_blue = np.array([110, 50, 50]) # h
upper_blue = np.array([130, 255, 255])
```

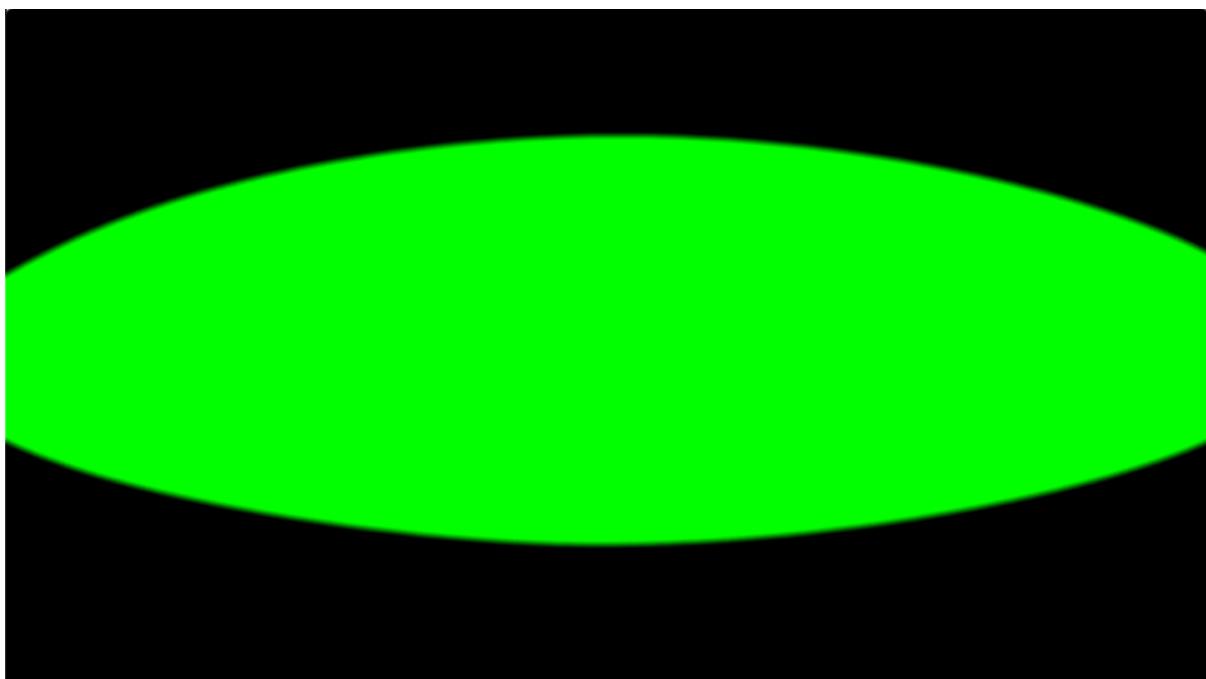
Change the blue range of HSV because blue, not green, must be removed.

Result

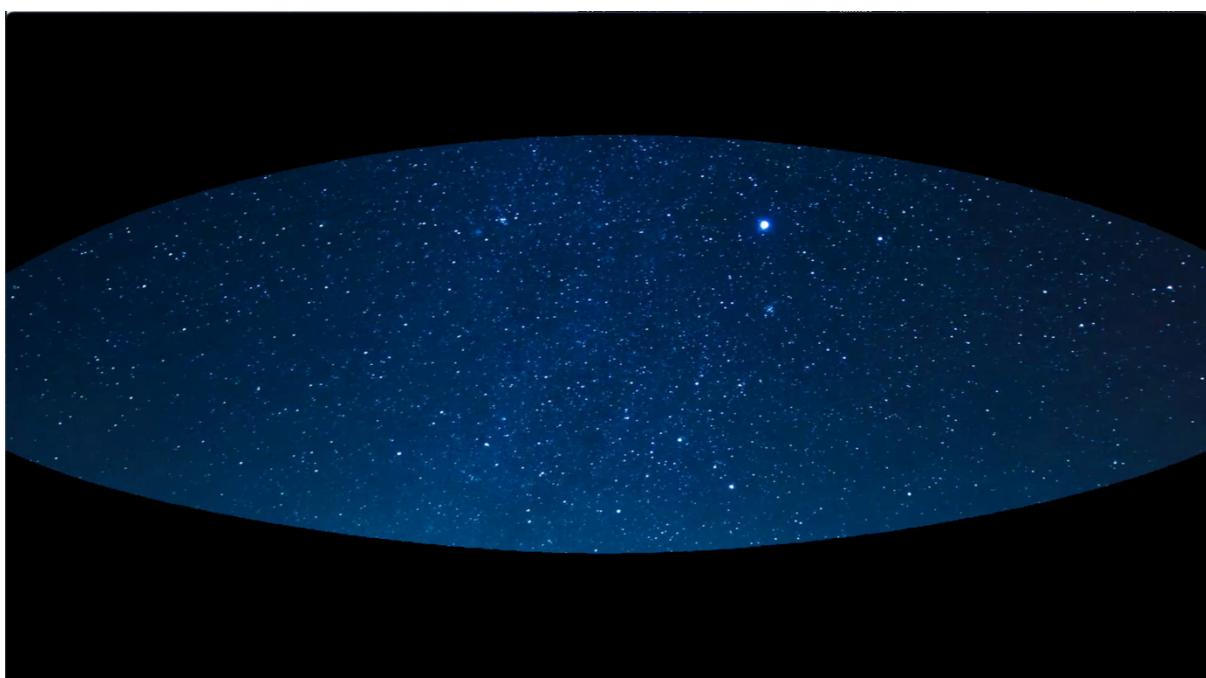


<3> Various effects sources with a green background Image taken in where the background scene is of a unique color

- Case3_(1) Blink

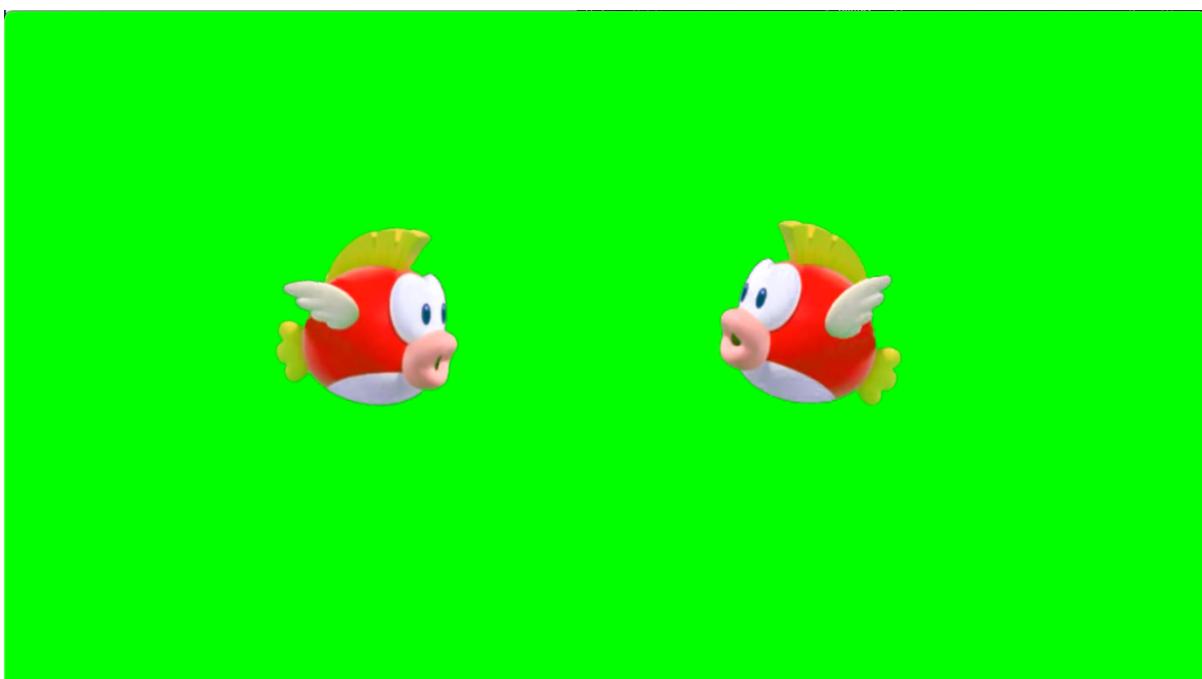


result

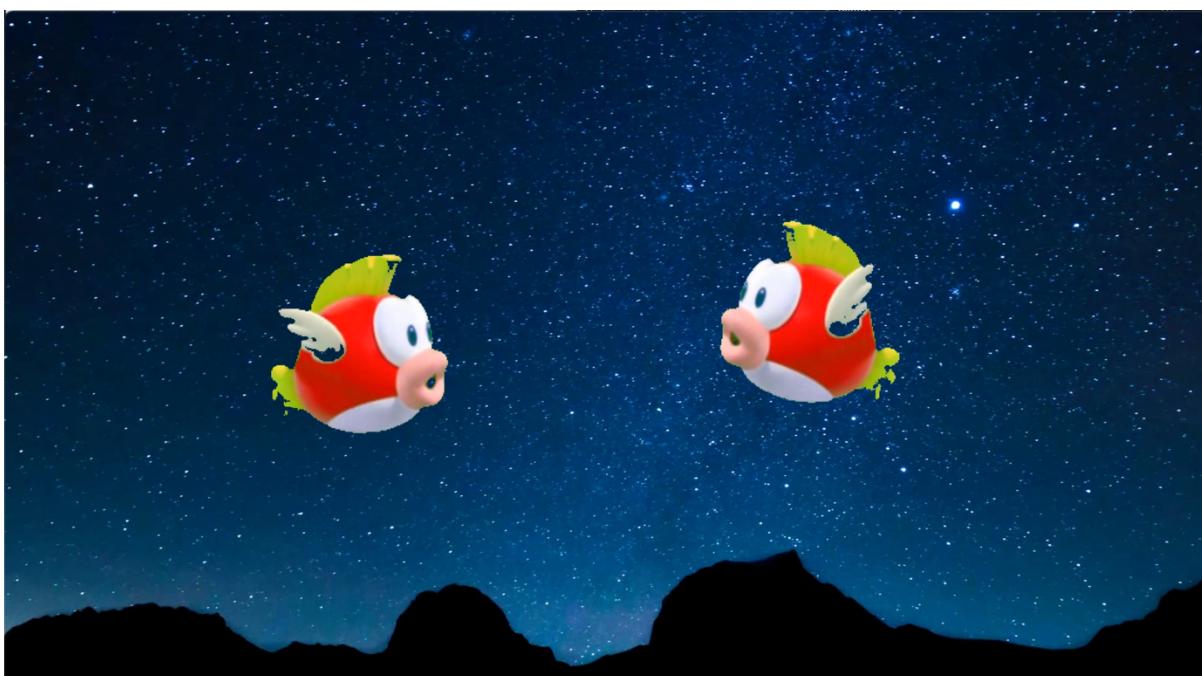


It can be confirmed that both images are detected clearly without additional code modification.

- Case3_(2) Fishes

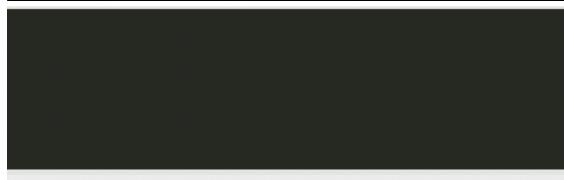


First Trial(lower_green = [40,40,40])



When applied to this video, there was a part where chroma key was not applied. So We searched the green range in HSV once again. When the range of lower green was changed from [40, 40, 40] to [50,100, 100], it was found that it was well applied.

Second Trial(lower_green = [50,100,100])



<input checked="" type="radio"/>	H	80	°
<input type="radio"/>	S	16	%
<input type="radio"/>	V	16	%

in [40, 40, 40], the lower_green color is like this.

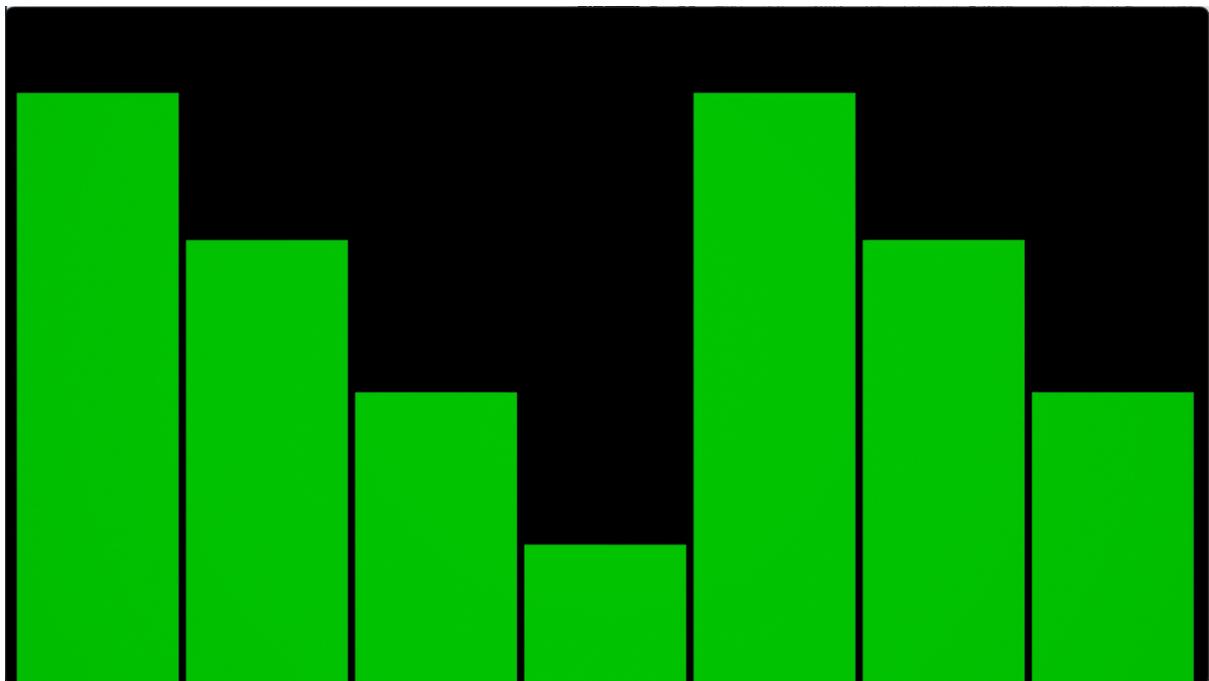


<input type="radio"/>	H	80	°
<input type="radio"/>	S	40	%
<input checked="" type="radio"/>	V	40	%

in [50,100,100], the lower_green color is like this.

In opencv, color range is different from the range of the picture because the range of hue is 180 degrees and the range of saturation and value is 256.

- Case3_(3) Slider



result



- Case3_(4) Snow effect



result



2. A revised matter

1. The function of recording images applied with chroma keys was added.
2. The range of Lower_green was changed from [40, 40, 40] to [50,100, 100].

3. What to update next

1. The phenomenon in which the code is played based on the background image, and the video ends when the background video ends not target video
 - ➔ We will find a way to modify it so that the video can be played based on the target video rather than the background video.
 - ➔ And also find a way to repeat the background video even after it ends.
 - ➔ We will find out what determines the length of the image and set it as a variable to modify it.

4. Code

```
1 import cv2 as cv
2 import numpy as np
3 import copy
4
5 width , height = 1920, 1080
6 video = cv.VideoCapture('/assets/case3/case3_fishes.mp4')
7 backv = cv.VideoCapture('/assets/sky.mp4')
8 recorder = cv.VideoWriter("recorder.mp4",
9                             cv.VideoWriter_fourcc(*'MP4V'),
10                            30,
11                            (width,height))
12 while(1):
13     # Take each frame
14     load_video, frames = video.read()
15     load_back, b_frames = backv.read()
16     frames = cv.resize(frames, (1920, 1080))
17     b_frames = cv.resize(b_frames, (1920, 1080))
18     # Convert BGR to HSV
19     hsv = cv.cvtColor(frames, cv.COLOR_BGR2HSV)
20
21     # define range of blue color in HSV
22     lower_green = np.array([50, 100, 100]) # hue(0-179), saturation(0-255), value(0-255)
23     upper_green = np.array([70, 255, 255])
24
25     # Threshold the HSV image to get only blue colors
26     mask = cv.inRange(hsv, lower_green, upper_green)
27     mask_v1 = cv.inRange(hsv, lower_green, upper_green)
28     # version 1: Use cv2.copyTo()
29     frames_v1 = copy.deepcopy(frames)
30     b_frames_v1 = copy.deepcopy(b_frames)
31     cv.copyTo(b_frames_v1,mask_v1,frames_v1)
32
33     cv.imshow('chromakey',frames_v1)
34     recorder.write(frames_v1)
35     if not load_video or not load_back:
36         break                                #video나 back 영상이 끝나면 break
37     if cv.waitKey(1) == 27:
38         break
39 backv.release()
40 video.release()
41 recorder.release()
42 cv.destroyAllWindows()
```