

GOAL



To use Color detection and segmentation techniques for detecting a specific color and making it “disappear”.

In simple words, this technique is opposite to the Green Screening technique. In Green Screening, we remove background but in this case we will be removing the foreground frame.



# ALGORITHM



- I. First capture the background frame and store it.
- II. Then detect red colored cloth using color detection and segmentation algorithm
- III. Generate a mask to segment out the red colored cloth
- IV. Finally, to achieve the finalized effect, generate the augmented output



# APPROACH



# LIBRARIES USED

- OpenCV - used for image processing aspect of the project



- Numpy - used to handle arrays as images



- Time



Reading, recording and caching the background for each frame. And giving camera some time to warm up.

We have to store the background image for each frame since we will be replacing the red colored pixels with the background pixels to create the invisible effect in the video.

```
17 capture_video = cv2.VideoCapture("video.mp4")
18
19 #give the camera to warm up
20 time.sleep(1)
21 count = 0
22 background = 0
23
24 #capturing the background in range of 60
25 for i in range(60):
26     return_val , background = capture_video.read()
27     if return_val == False :
28         continue
29
30 background = np.flip(background, axis=1)
31
```



## Detecting red color portion In each frame

In this step, now we will focus on the red colored parts. First, we will convert RGB (red-blue-green) to HSV (hue-saturation-value) since RGB values are highly sensitive to illumination.

Then, we will specify the range of color to detect red color in the video.

```
32 # we are reading from video
33 while (capture_video.isOpened()):
34     return_val, img = capture_video.read()
35     if not return_val :
36         break
37     count = count + 1
38     img = np.flip(img , axis=1)
39     # convert the image - BGR to HSV
40     # as we focused on detection of red color
41     hsv = cv2.cvtColor(img , cv2.COLOR_BGR2HSV)
42     # generating mask to detect red color
43     # HSV
44     # it should be mono-color cloth
45     # lower range
46     lower_red = np.array([100, 40, 40])
47     upper_red = np.array([100, 255, 255])
48     mask1 = cv2.inRange(hsv,lower_red,upper_red)
49
50     lower_red = np.array([155, 40, 40])
51     upper_red = np.array([180, 255, 255])
52     mask2 = cv2.inRange(hsv,lower_red,upper_red)
53
54     mask1 = mask1+mask2
55
```





## Replacing the red portion with a mask image in each frame

Now, we have a red part of the video in the 'mask' image, we will segment the mask part from the frames and we will do a morphology open and dilation for that.

```
55
56     # Refining the mask corresponding to the detected red color
57     mask1 = cv2.morphologyEx(mask1, cv2.MORPH_OPEN, np.ones((3,3),np.uint8),iterations=2)
58     mask1 = cv2.dilate(mask1,np.ones((3,3),np.uint8),iterations = 1)
59     mask2 = cv2.bitwise_not(mask1)
60
```



# Generating the final output

```
60
61     # Generating the final output
62     res1 = cv2.bitwise_and(background,background,mask=mask1)
63     res2 = cv2.bitwise_and(img,img,mask=mask2)
64     final_output = cv2.addWeighted(res1,1,res2,1,0)
65
66     cv2.imshow("Disappearing red color",final_output)
67     k = cv2.waitKey(10)
68     if k == 27:
69         break
70
71     capture_video.release()
72     cv2.destroyAllWindows()
```



# RESULTS





Demo Input Video



Demo Output Video



*Thank  
You*

