Abstract

Have you ever thought of making visible things invisible, just like the Harry Potter? Have you ever thought how does one supersede backgrounds and add effects in a movie? The cloak was magical and invisible in Harry Potter movie. As we know there is no magic and no invisible cloak which exists in the world. It’s all about the graphics tricks. The concept of an invisibility cloak is a mixture of science, fantasy, and the collective imagination. This helps to create one’s own ‘Invisibility Cloak. It will make use of Python and OpenCV module specifically targeting Image Processing and Image Segmentation to create a false sense of invisibility in the frame. It will explore how an object of a specific color or texture can be manipulated using the OpenCV library of python. To achieve this, initially we’ll be capturing and storing the backdrop frame. Thereafter we’ll be identifying the given-colored fabric by making use of the above-mentioned algorithms. Then we’ll segment out the given-colored fabric by generating a mask and then finally, we’ll generate the final augmented(magical) output to create Invisibility cloak.

I. Methodology

Steps:

1. Capture the background frame (it will take few second to read it).

The main concept is interchanging the current frame picture element equivalent to the cloth with the backdrop pixels, so we can obtain the magical effect of invisibility, Thus it required to save the frame in the background.

2. Identify the chose colored cloth (cloak) by using the color detection.

3. Than use deep learning algorithm for the color and image segmentation algorithm.

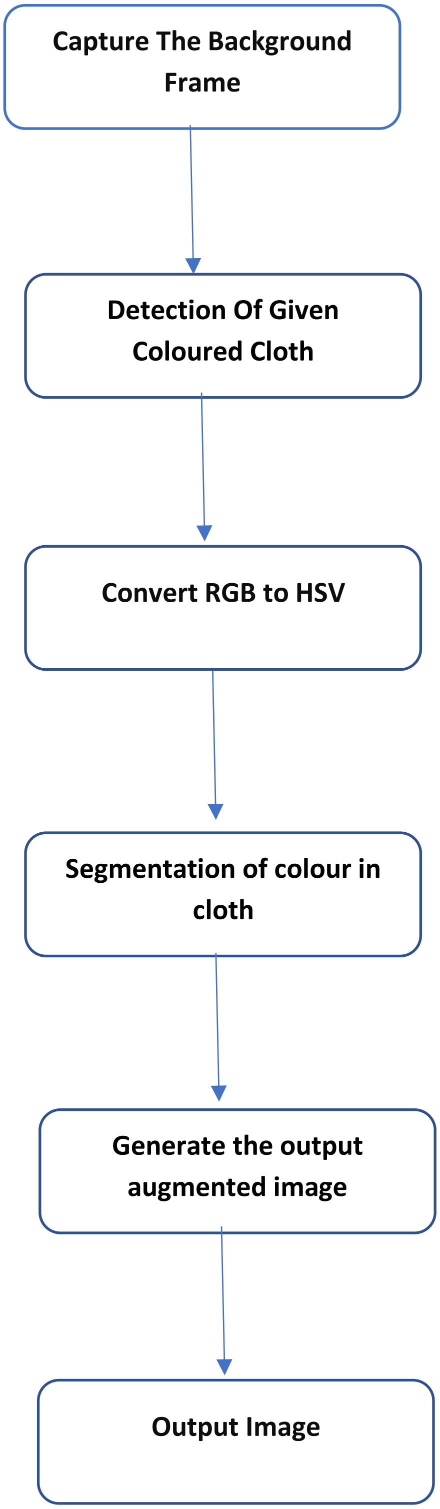
4. The correct concept is to change the color-space of the picture from 'R.G.B (Red. Green. Blue) value to 'H.S.V' (Hue . Saturation . Value).

Since, the "R.G.B values" are highly - 'sensitive to Illumination. Thus ,the proper way, is transforming the color space of our picture from R.G.B to H.S.V (Hue .Saturation .Value). ["mask1 = mask1 + mask2"]
Using this, we join the masks generated for both of the given color range.

5. Segmenting out the red colored fabric by generating a mask. We refine the mask & then it is further used for segmenting out the fabric from the frame.

6. Generate the final augmented(magical) output to create
Invisibility cloak.

II.1 Flow Chart :


Capture The Background Frame
Detection Of Given Coloured Cloth
Convert RGB to HSV
Segmentation of colour in cloth
Generate the output augmented image
Output Image

II. Output

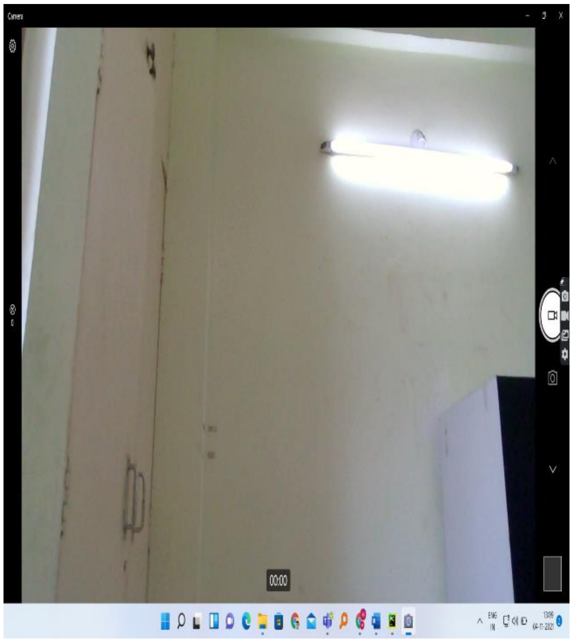


Fig 3.2 Captured Background image

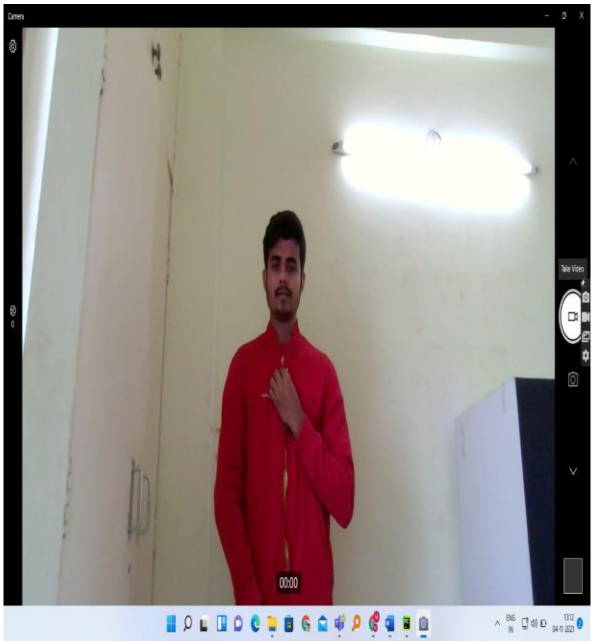


Fig 3.3 Input frames

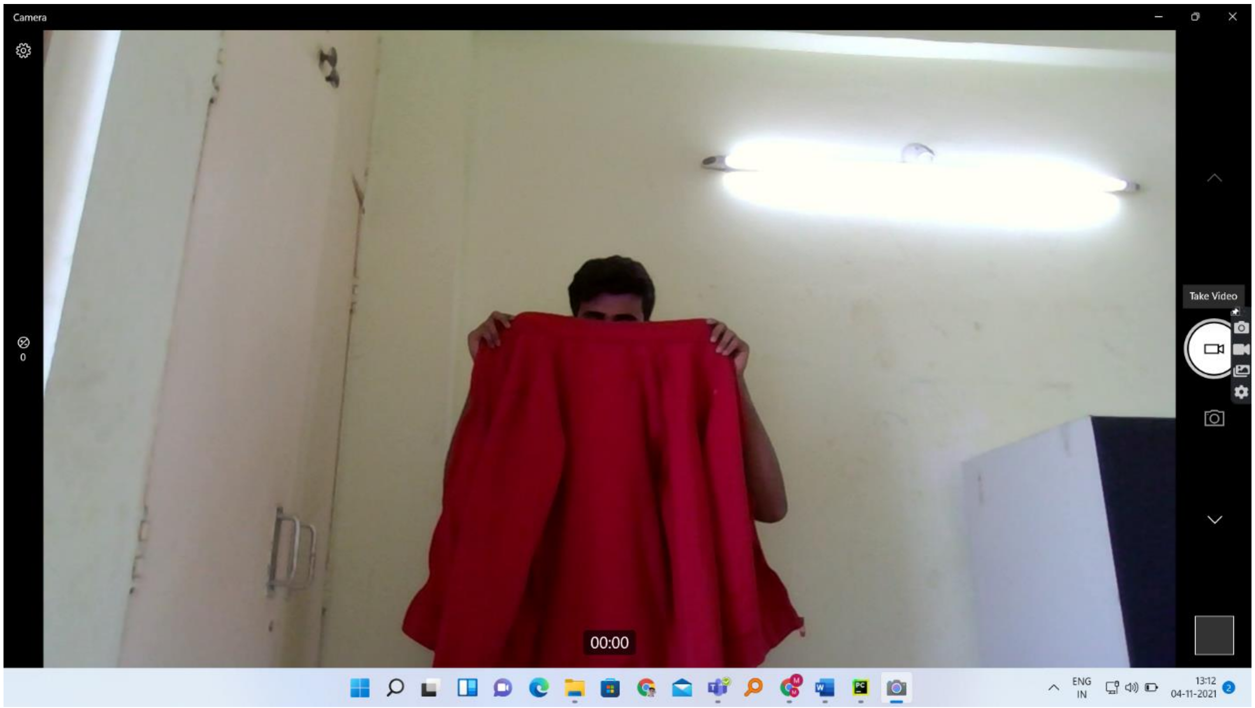


Fig 3.3 Input

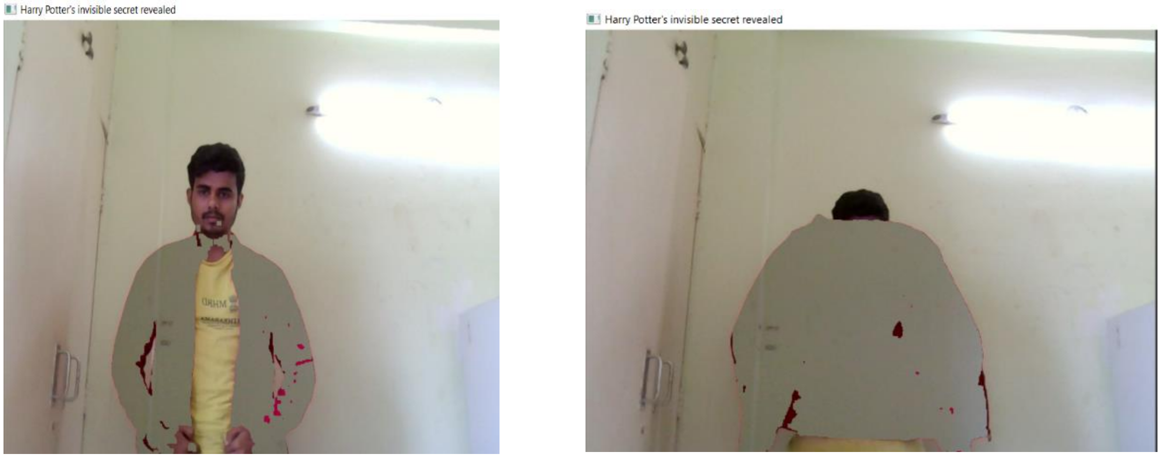

Harry Potter's invisible secret revealed
Harry Potter's invisible secret revealed

Fig 3.5 Output (realtime)

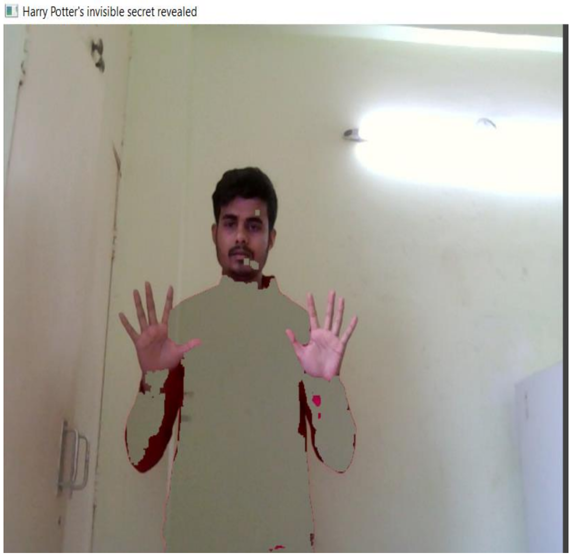


Fig 3.6 Final output

III. Discussion

Our main task is ‘to substitute the current frame pixels corresponding to the fabric with the backdrop pixels to create the effect of an invisibility-cloak and we implemented it successfully. Computer vision has still not achieved a level wherein it can be directly put into use to solve life problems, as it is still in its developmental phase. Also, it can be widely used in the applications of Augmented Reality.

Some of the applications are
• Video Editing & media in order to create stunning visuals.
• "Infinity Tower" is an invisible skyscraper in South Korea (under construction).

IV. Conclusion

Computer vision can be used to solve the most problems with utmost sophistication. All the basics regarding the colour detection technique along with different ways to achieve it have been in this report . During the course of programming, we can useboth Python and MATLAB for Computer Vision, but I prefer Python because it takes less simulation time than MATLAB.

V. References

(Paper Implemented)

https://www.ijeat.org/wp-content/uploads/papers/v9i4/D7531049420.pdf

Appendix:

```
import numpy as np
import cv2
import time

cap = cv2.VideoCapture(0)  # Read from the web cam

time.sleep(3)  # for the system to sleep for 3 second before the webcam
starts
for i in range(30):
    retval, back = cap.read()
back = np.flip(back, axis=1)
cap = cv2.VideoCapture(0)

## detecting the red portion In each frame


while (cap.isOpened()):  ##Read every Frame from the webcam, until the camera
is open
    ret, img = cap.read()
    if ret:
        img = np.flip(img, axis=1)

        ##convert the color space from BGR to HSV
        hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

        ##Generat masks to detect red color
        lower_red = np.array([0, 120, 70])
        upper_red = np.array([10, 255, 255])
        mask1 = cv2.inRange(hsv, lower_red, upper_red)

        lower_red = np.array([170, 120, 70])
        upper_red = np.array([180, 255, 255])
        mask2 = cv2.inRange(hsv, lower_red, upper_red)
        mask1 += mask2

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

        mask = cv2.morphologyEx(mask1, cv2.MORPH_OPEN, np.ones((5, 5),
np.uint8))
        img[np.where(mask == 255)] = back[np.where(mask == 255)]

        # Final output
        cv2.imshow("Harry Potter's invisible secret revealed", img)
    key = cv2.waitKey(1)
    if key == ord("q"):
        break
cap.release()
cv2.destroyAllWindows()
```