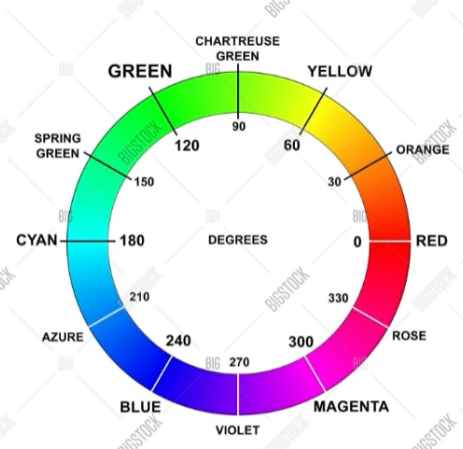
**INVISIBLE CLOAK**



import pandas as pd

import numpy as np

import cv2

import time # to provide the camera sum time before tthe code is executed

# create a video capture object

cap = cv2.VideoCapture(0) # 0 is for inbuilt webcam & 1 is for external connected webcam

time.sleep(2) # 2 seconds sleep :- Whenever webcamera start it takes time to set hence we have give 2 seconds

# time for adjustmenmts

# what background need to be show when we have the cloak on ourself :- 0 represent invisibility

background = 0

# Running the for loop 30 times with 30 iterations

# before coming in front of the webcam , allowing the web cam to capture the background

for i in range(30):

ret, background = cap.read() # Here i am capturing the background

# cap.read() returns two things i.e the image that is captured

# and the return value that is true

# Till the capture for object is running this while loop will be executing

while(cap.isOpened()):

ret, img = cap.read() # Here i am capturing the my image to perform operation on it

if not ret:

break

# Converting from BGR to HSV

hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV) #HSB or HSV = Hue saturation value / brightness

lower\_red = np.array([0,120,70])

upper\_red = np.array([10,255,255])

mask1 = cv2.inRange(hsv, lower\_red, upper\_red) # Separating the Cloak part

lower\_red = np.array([170,120,70])

upper\_red = np.array([180,255,255])

mask2 = cv2.inRange(hsv, lower\_red, upper\_red)

mask1 = mask1 + mask2 # OR 1 or x

mask1 = cv2.morphologyEx(mask1, cv2.MORPH\_OPEN,

np.ones((3,3),np.uint8), iterations=2) #Noise Removal

mask1 = cv2.morphologyEx(mask1, cv2.MORPH\_DILATE,

np.ones((3,3),np.uint8), iterations=1) #Smoothness of the image

mask2 = cv2.bitwise\_not(mask1) # except the claok eveything would be there

res1 = cv2.bitwise\_and(background,background,mask=mask1) # Used for segmentation of the color

res2 = cv2.bitwise\_and(img, img, mask=mask2) #Used to substitute the cloak part

final\_output = cv2.addWeighted(res1, 1, res2, 1, 0)

cv2.imshow('Eureka !!',final\_output)

k=cv2.waitkey(10)

if k == 27 :

break

cap.release()

cv2.destroyAllWindows()

**87687687687**

import cv2

import numpy as np

import time

print("""

Harry : Hey !! Would you like to try my invisibility cloak ??

Its awesome !!

Prepare to get invisible .....................

""")

cap = cv2.VideoCapture(0)

time.sleep(3)

background=0

for i in range(30):

ret,background = cap.read()

background = np.flip(background,axis=1)

while(cap.isOpened()):

ret, img = cap.read()

# Flipping the image (Can be uncommented if needed)

img = np.flip(img,axis=1)

# Converting image to HSV color space.

hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)

value = (35, 35)

blurred = cv2.GaussianBlur(hsv, value,0)

# Defining lower range for red color detection.

lower\_red = np.array([0,120,70])

upper\_red = np.array([10,255,255])

mask1 = cv2.inRange(hsv,lower\_red,upper\_red)

# Defining upper range for red color detection

lower\_red = np.array([170,120,70])

upper\_red = np.array([180,255,255])

mask2 = cv2.inRange(hsv,lower\_red,upper\_red)

# Addition of the two masks to generate the final mask.

mask = mask1+mask2

mask = cv2.morphologyEx(mask, cv2.MORPH\_OPEN, np.ones((5,5),np.uint8))

# Replacing pixels corresponding to cloak with the background pixels.

img[np.where(mask==255)] = background[np.where(mask==255)]

cv2.imshow('Display',img)

k = cv2.waitKey(10)

if k == 27:

break

**87687687687**

## How it works

#1. Importing needed libraries and generate the output video

#2. Recording and caching the background for each frame.

#3. detecting the red portion in each frame

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

#5. Producing the surprising output

#Importing libraries

import numpy as np

import cv2

import time

#Recording and caching the background for each frame.

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()