PRO-C121

Capstone class : Invisibility Cloak

What is our GOAL for this MODULE?

The goal of this module is to explore the image processing techniques such as saturation and segmentation.

What did we ACHIEVE in the class TODAY?

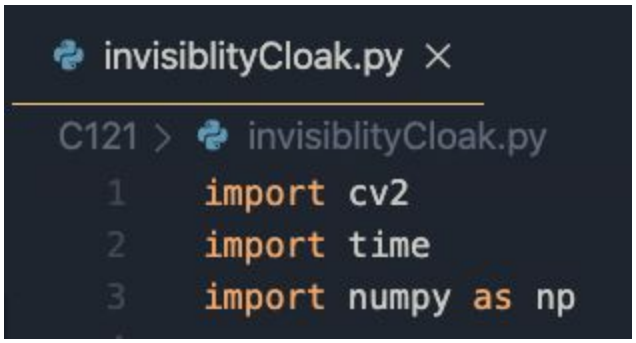
- We wrote an algorithm to create an invisibility cloak.

Which CONCEPTS/CODING BLOCKS did we cover today?

- CV2 (Camera vision library)
- Saturation and segmentation techniques

How did we DO the activities?

1. We imported the cv2 , time and numpy libraries to our file.



2. We wrote code and prepared for writing the output video.

```
#To save the output in a file output.avi
fourcc = cv2.VideoWriter_fourcc(*'XVID')
output_file = cv2.VideoWriter('output.avi', fourcc, 20.0, (640, 480))
```

3. We started to read the video and warmed up the camera.

```
#Starting the webcam
cap = cv2.VideoCapture(0)

#Allowing the webcam to start by making the code sleep for 2 seconds
time.sleep(2)
bg = 0
```

4. We coded to capture the background in range of 60.

```
#Capturing background for 60 frames
for i in range(60):
    ret, bg = cap.read()
#Flipping the background
bg = np.flip(bg, axis=1)
```

5. We read every frame from the webcam, until the camera is open.

```
#Reading the captured frame until the camera is open
while (cap.isOpened()):
    ret, img = cap.read()
    if not ret:
        break
    #Flipping the image for consistency
    img = np.flip(img, axis=1)
```

6. We coded to convert the color from BGR to HSV for better detection.

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

7. Then we coded to create the masks.

```
#Generating mask to detect red colour
#These values can also be changed as per the color
lower_red = np.array([0, 120, 50])
upper_red = np.array([10, 255,255])
mask_1 = cv2.inRange(hsv, lower_red, upper_red)

lower_red = np.array([170, 120, 70])
upper_red = np.array([180, 255, 255])
mask_2 = cv2.inRange(hsv, lower_red, upper_red)

mask_1 = mask_1 + mask_2

#Open and expand the image where there is mask 1 (color)
mask_1 = cv2.morphologyEx(mask_1, cv2.MORPH_OPEN, np.ones((3, 3), np.uint8))
mask_1 = cv2.morphologyEx(mask_1, cv2.MORPH_DILATE, np.ones((3, 3), np.uint8))
```

8. Then we created an inverted mask to segment out the red color from the frame.

```
#Selecting only the part that does not have mask one and saving in mask 2
mask_2 = cv2.bitwise_not(mask_1)
```

9. We then segmented the red color part out of the frame using bitwise and inverted mask.

```
#Keeping only the part of the images without the red color
#(or any other color you may choose)
res_1 = cv2.bitwise_and(img, img, mask=mask_2)
```

10. We wrote code to create an image showing static background frame pixels only for the masked region.

```
#Keeping only the part of the images with the red color
#(or any other color you may choose)
res_2 = cv2.bitwise_and(bg, bg, mask=mask_1)
```

11. We coded to get the video as our output and close all the windows.

```
    #Generating the final output by merging res_1 and res_2
    final_output = cv2.addWeighted(res_1, 1, res_2, 1, 0)
    output_file.write(final_output)
    #Displaying the output to the user
    cv2.imshow("magic", final_output)
    cv2.waitKey(1)


cap.release()
out.release()
cv2.destroyAllWindows()
```

What's NEXT?

In the next class, we will learn about neural networks.

EXTEND YOUR KNOWLEDGE:

Learn more about the cv2 library from the following link:
https://docs.opencv.org/master/d6/d00/tutorial_py_root.html