

임베디드응용및실습 과제

10주차. 과제 보고서

과 목 명	임베디드응용및실습
학 번	2020161092
이 름	이재윤
제 출 일	2024년11월09일

1-1 실습문제

OpenCV를 사용하여 라즈베리파이 카메라에서 받은 실시간 영상으로 얼굴 검출

1-2. 소스코드

```
import numpy as np
import cv2

face_cascade = cv2.CascadeClassifier("data/haarcascades/haarcascade_frontalface_default.xml")

cap = cv2.VideoCapture(0, cv2.CAP_V4L) # 첫번째 카메라 영상
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640) # 카메라 영상 넓이
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480) # 카메라 영상 높이

while True:
    ret, img = cap.read()
    img = cv2.flip(img, -1)

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.2, 5)
    print("Number of faces detected:", len(faces))

    for (x, y, w, h) in faces:
        img = cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 1)

    cv2.imshow('img', img)

    k = cv2.waitKey(30) & 0xff
    if k == 27: # Esc 키를 누르면 종료
        break

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

1-3. 실행결과(+영상으로 제출)



2-1 실습문제

첨부된 4장의 이미지를 라인 트레이서 용도로 얻었다고 가정하고, 4장의 영상에서 노란색 또는 흰색선을 추출하여 표기하시오.

- 영상 표기하는 방법은 자유롭게 한다
- 사각형, 라인, 선만 남기고 다 검게 등등...
- 제안 알고리즘을 4장의 영상에 동일하게 적용 시, 성능이 보장되도록 한다
- 영상 크기 변경, 크롭, 컬러 변경 등 자유롭게 할 수 있다 • 소스 코드와 라인 표기된 4장의 영상 제출

2-2. 소스코드

```
import cv2
import numpy as np

img = cv2.imread("./1.jpg", cv2.IMREAD_COLOR) # 1.jpg, 2.jpg, 3.jpg, 4.jpg로만 바꿔서 사용

if img is not None:
    img = cv2.resize(img, (400, 300))
    print('img shape:', img.shape)

    lower_yellow = np.array([0, 100, 100]) # 노란색 범위 설정
    upper_yellow = np.array([50, 255, 255]) # 약간의 오차허용을 위해 블루 채널에 50 추가
    yellow_mask = cv2.inRange(img, lower_yellow, upper_yellow) # lower_yellow에서
upper_yellow 사이의 값을 마스크로 추출(이 외 값은 제외)
    yellow_result = cv2.bitwise_and(img, img, mask=yellow_mask) # img에서 yellow_mask인 영역만 살림

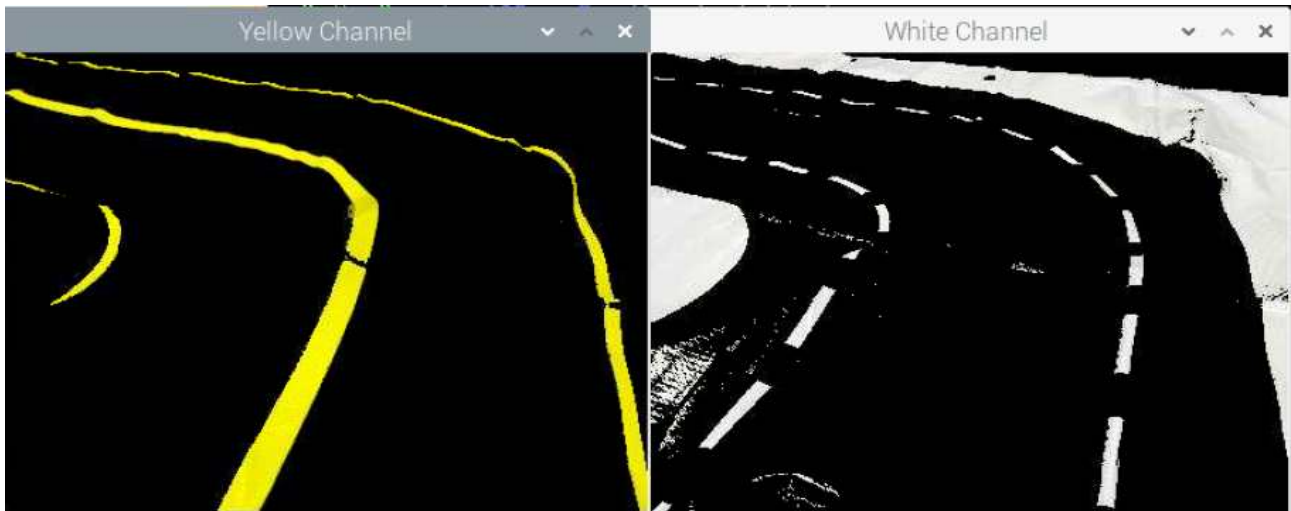
    lower_white = np.array([200, 200, 200]) # 흰색 범위 설정
    upper_white = np.array([255, 255, 255])
    white_mask = cv2.inRange(img, lower_white, upper_white)
    white_result = cv2.bitwise_and(img, img, mask=white_mask) # img에서 white_mask인 영역만 살림

    cv2.imshow("Yellow Channel", yellow_result) # 노란색만 추출
    cv2.imshow("White Channel", white_result) # 흰색만 추출

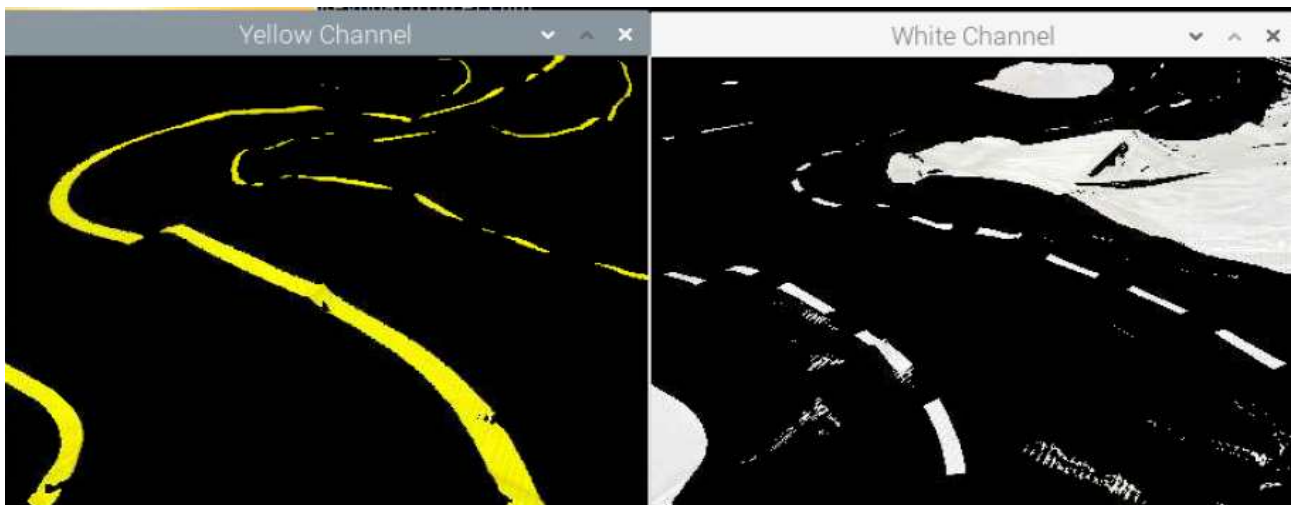
    cv2.waitKey(0)
    cv2.destroyAllWindows()
else:
    print("Image file not found")
```

2-3 실행결과(+사진 첨부)

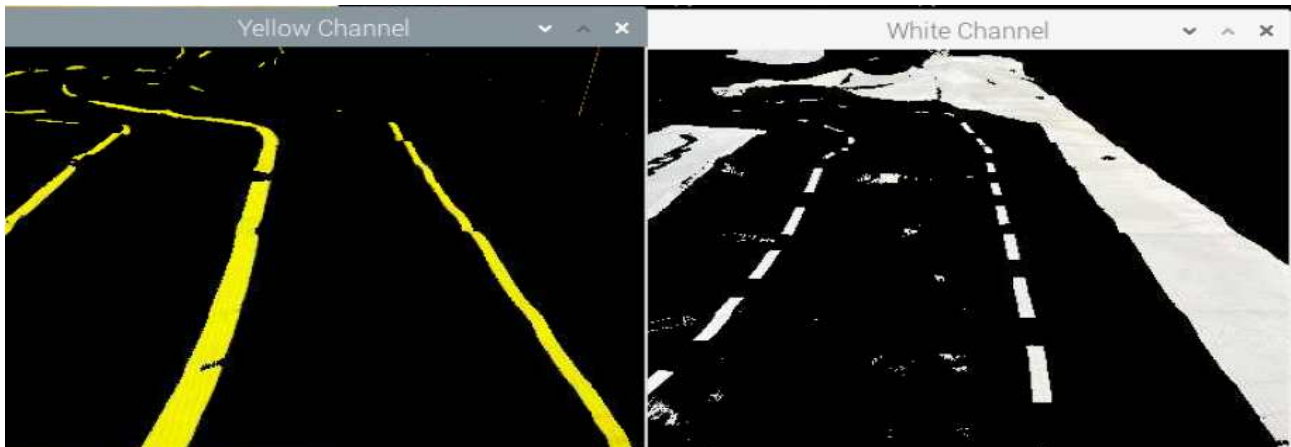
1.jpg



2.jpg



3.jpg



4.jpg

