

임베디드 과제

10주차. 과제 보고서

과 목 명	임베디드
학 번	2021161002
이 름	강민구
제 출 일	2025년11월07일

1-1 실습문제

OpenCV를 사용하여 라즈베리파이 카메라에서 받은 실시간 영상으로 얼굴 검출

1-2. 소스코드

```
import cv2
import numpy as np

xml_path = cv2.data.haarcascades + 'haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(xml_path)

threshold_move = 50
diff_compare = 10

cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

while True:
    _, img = cap.read()
    #img_third = cv2.flip(img_third, 0)
    scr = img.copy()
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces_gray = face_cascade.detectMultiScale(img_gray)

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

    cv2.imshow('Motion detection', scr)

    if cv2.waitKey(1) & 0xFF == 27:
        break

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

2-1 실습문제

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

2-2. 소스코드

```
import cv2 as cv
```

```
img1 = cv.imread('./imgs/1.jpg')
```

```
img2 = cv.imread('./imgs/2.jpg')
```

```
img3 = cv.imread('./imgs/3.jpg')
```

```
img4 = cv.imread('./imgs/4.jpg')
```

```
#img1
```

```
cv.imshow('img1',img1)
```

```
cv.waitKey(0)
```

```
_, mask1 = cv.threshold(img1, 30, 255, cv.THRESH_BINARY)
```

```
cv.imshow('mask1',mask1)
```

```
cv.waitKey(0)
```

```
mask1gray = cv.cvtColor(mask1,cv.COLOR_BGR2GRAY)
```

```
cv.imshow('mask1gray',mask1gray)
```

```
cv.waitKey(0)
```

```
_, mask1_th = cv.threshold(mask1gray, 240, 255, cv.THRESH_BINARY)
```

```
cv.imshow('mask1_th',mask1_th)
```

```
cv.waitKey(0)
```

```
mask1_inv = cv.bitwise_not(mask1_th)
```

```
cv.imshow('mask1_inv',mask1_inv)
```

```
cv.waitKey(0)
```

```
#img2
```

```
cv.imshow('img2',img2)
```

```
cv.waitKey(0)
```

```
_, mask2 = cv.threshold(img2, 30, 255, cv.THRESH_BINARY)
```

```
cv.imshow('mask2',mask2)
```

```
cv.waitKey(0)
```

```
mask2gray = cv.cvtColor(mask2,cv.COLOR_BGR2GRAY)
cv.imshow('mask2gray',mask2gray)
cv.waitKey(0)
```

```
_, mask2_th = cv.threshold(mask2gray, 240, 255, cv.THRESH_BINARY)
cv.imshow('mask2_th',mask2_th)
cv.waitKey(0)
```

```
mask2_inv = cv.bitwise_not(mask2_th)
cv.imshow('mask2_inv',mask2_inv)
cv.waitKey(0)
```

```
#img3
cv.imshow('img3',img3)
cv.waitKey(0)
```

```
_, mask3 = cv.threshold(img3, 30, 255, cv.THRESH_BINARY)
cv.imshow('mask3',mask3)
cv.waitKey(0)
```

```
mask3gray = cv.cvtColor(mask3,cv.COLOR_BGR2GRAY)
cv.imshow('mask3gray',mask3gray)
cv.waitKey(0)
```

```
_, mask3_th = cv.threshold(mask3gray, 240, 255, cv.THRESH_BINARY)
cv.imshow('mask3_th',mask3_th)
cv.waitKey(0)
```

```
mask3_inv = cv.bitwise_not(mask3_th)
cv.imshow('mask3_inv',mask3_inv)
cv.waitKey(0)
```

```
#img4
cv.imshow('img4',img4)
cv.waitKey(0)
```

```
_, mask4 = cv.threshold(img4, 30, 255, cv.THRESH_BINARY)
cv.imshow('mask4',mask4)
```

```
cv.waitKey(0)
```

```
mask4gray = cv.cvtColor(mask4,cv.COLOR_BGR2GRAY)
```

```
cv.imshow('mask4gray',mask4gray)
```

```
cv.waitKey(0)
```

```
_, mask4_th = cv.threshold(mask4gray, 240, 255, cv.THRESH_BINARY)
```

```
cv.imshow('mask4_th',mask4_th)
```

```
cv.waitKey(0)
```

```
mask4_inv = cv.bitwise_not(mask4_th)
```

```
cv.imshow('mask4_inv',mask4_inv)
```

```
cv.waitKey(0)
```

2-3. 실행결과



