# 임베디드 응용 및 실습 과제6

- 6. OpenCV


| | |
|---|---|
| 학과 | 전기공학과 |
| 학년 | 3학년 |
| 학번 | 2020161110 |
| 이름 | 전현서 |
| 과목명 | 임베디드 응용 및 실습 |
| 분반 | 01분반 |
| 담당교수님 | 우성민 교수님 |

# 1. 얼굴인식

```
import numpy as np
import cv2
face_cascade =cv2.CascadeClassifier('/home/wjsgustj/week10/assignment/haarcascade_frontalface_default.xml')
eye_cascade =cv2.CascadeClassifier('/home/wjsgustj/week10/assignment/haarcascade_eye.xml')
cap =cv2.VideoCapture(0, cv2.CAP_V4L)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
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)
    for(x, y, w, h) in faces:
        img =cv2.rectangle(img, (x, y), (x +w, y +h), (255, 0, 0), 1)
        roi_gray =gray[y:y +h, x:x +w]
        roi_color =img[y:y +h, x:x +w]
    cv2.imshow('img', img)
    k =cv2.waitKey(30) &0xff


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

## 2. 라인인식

```
import cv2
import numpy as np
def extract_white_yellow_with_refined_boundaries(image_path):
    image =cv2.imread(image_path)
    height, width, _=image.shape

    # 이미지를 HSV 색상 공간으로 변환
    hsv =cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    # 흰색 범위 설정 (HSV)
    lower_white =np.array([0, 0, 200])
    upper_white =np.array([180, 50, 255])

    # 노란색 범위 설정 (HSV)
    lower_yellow =np.array([23, 140, 140])
    upper_yellow =np.array([30, 255, 255])

    # 노란색 마스크 생성
    mask_yellow =cv2.inRange(hsv, lower_yellow, upper_yellow)

    # 각 높이별로 노란색 선의 좌우 경계 찾기
    left_bound =np.zeros(height, dtype=int)
    right_bound =np.full(height, width -1, dtype=int)

    for y in range(height):
        yellow_indices =np.where(mask_yellow[y, :] >0)[0]
        if len(yellow_indices) >0:
            left_bound[y] =yellow_indices[0]
            right_bound[y] =yellow_indices[-1]
    # 흰색 마스크 생성
    mask_white =cv2.inRange(hsv, lower_white, upper_white)

    # 흰색 마스크를 노란색 좌우 경계 안쪽으로 제한
    for y in range(height):
        mask_white[y, :left_bound[y]] =0
        mask_white[y, right_bound[y]:] =0
    # 흰색과 노란색 마스크를 결합
    mask_combined =cv2.bitwise_or(mask_white, mask_yellow)

    # 원본 이미지에서 도로 내부의 흰색과 노란색 부분만 추출
    result =cv2.bitwise_and(image, image, mask=mask_combined)

    # 결과 출력 (OpenCV를 사용하여 화면에 표시)
    cv2.imshow("Extracted White and Yellow", result)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
# 4개의 이미지 파일에 대해 절대 경로로 실행
```

```
extract_white_yellow_with_refined_boundaries('/home/wjsgustj/week10/1.jpg')
extract_white_yellow_with_refined_boundaries('/home/wjsgustj/week10/2.jpg')
extract_white_yellow_with_refined_boundaries('/home/wjsgustj/week10/3.jpg')
extract_white_yellow_with_refined_boundaries('/home/wjsgustj/week10/4.jpg')
```