얼굴 탐지(Face Detection)

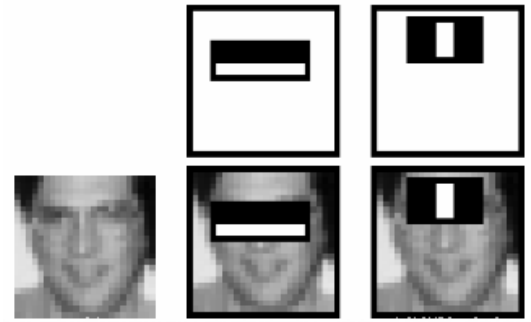
박경규

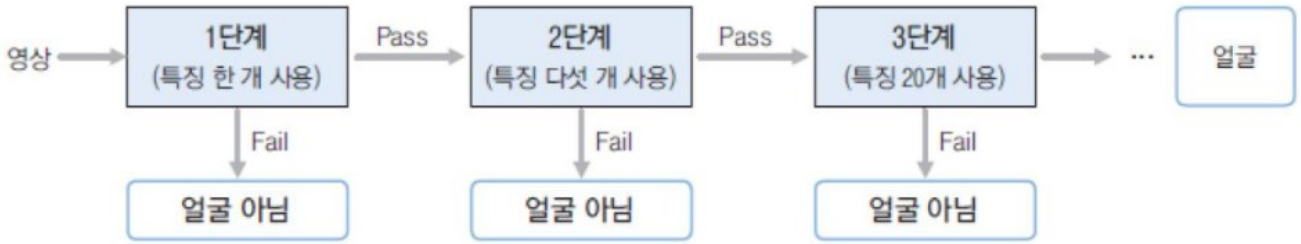
CASCADE CLASSIFIER

(a) Edge Features

(b) Line Features

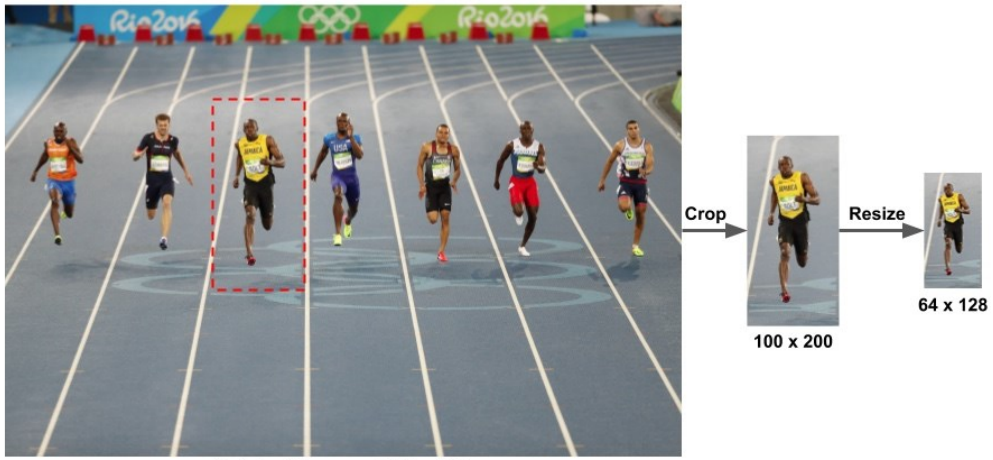
(c) Four-rectangle features



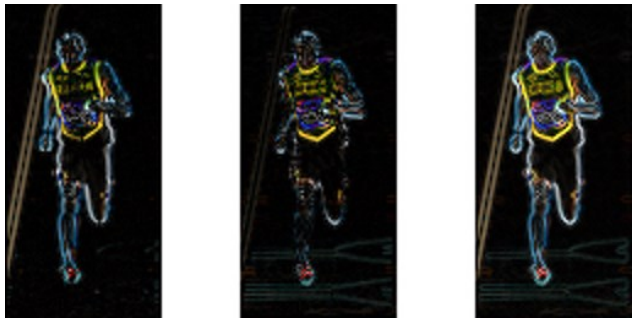


HOG - Histograms of Oriented Gradients

Step 1 : Preprocessing

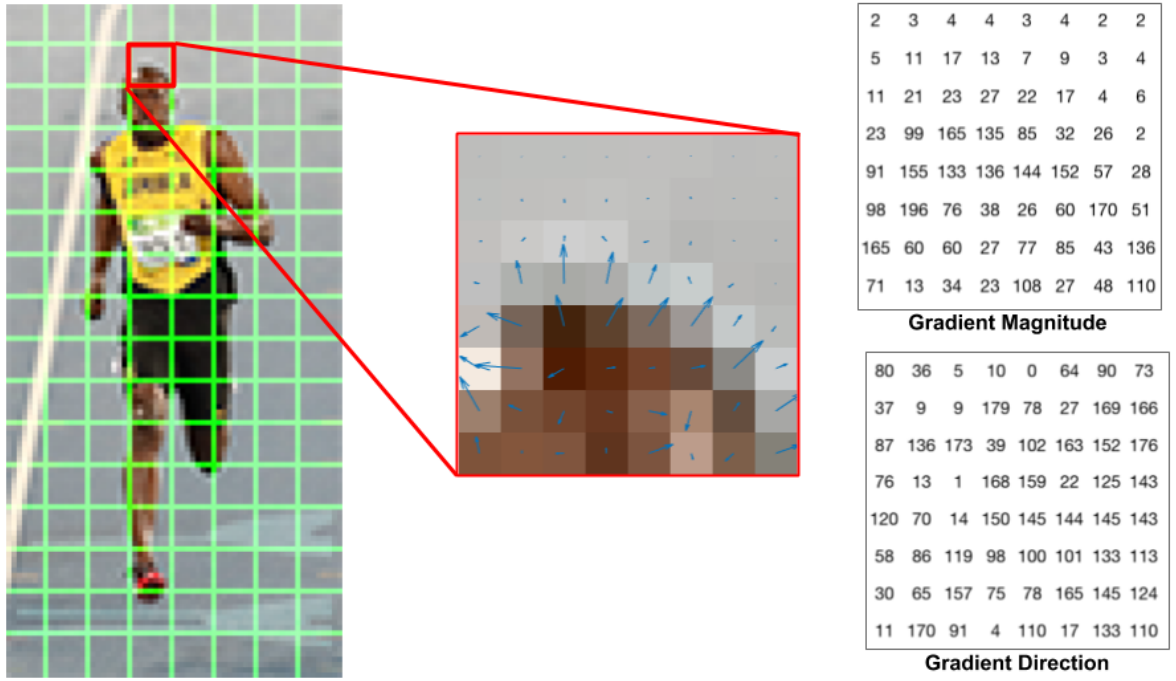


Step 2 : Calculate the Gradient Images



왼쪽: X-gradient의 절대값
가운데 : y-gradient의 절대값
오른쪽 : gradient의 크기.

Step 3 : Calculate Histogram of Gradients in 8×8 cells



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

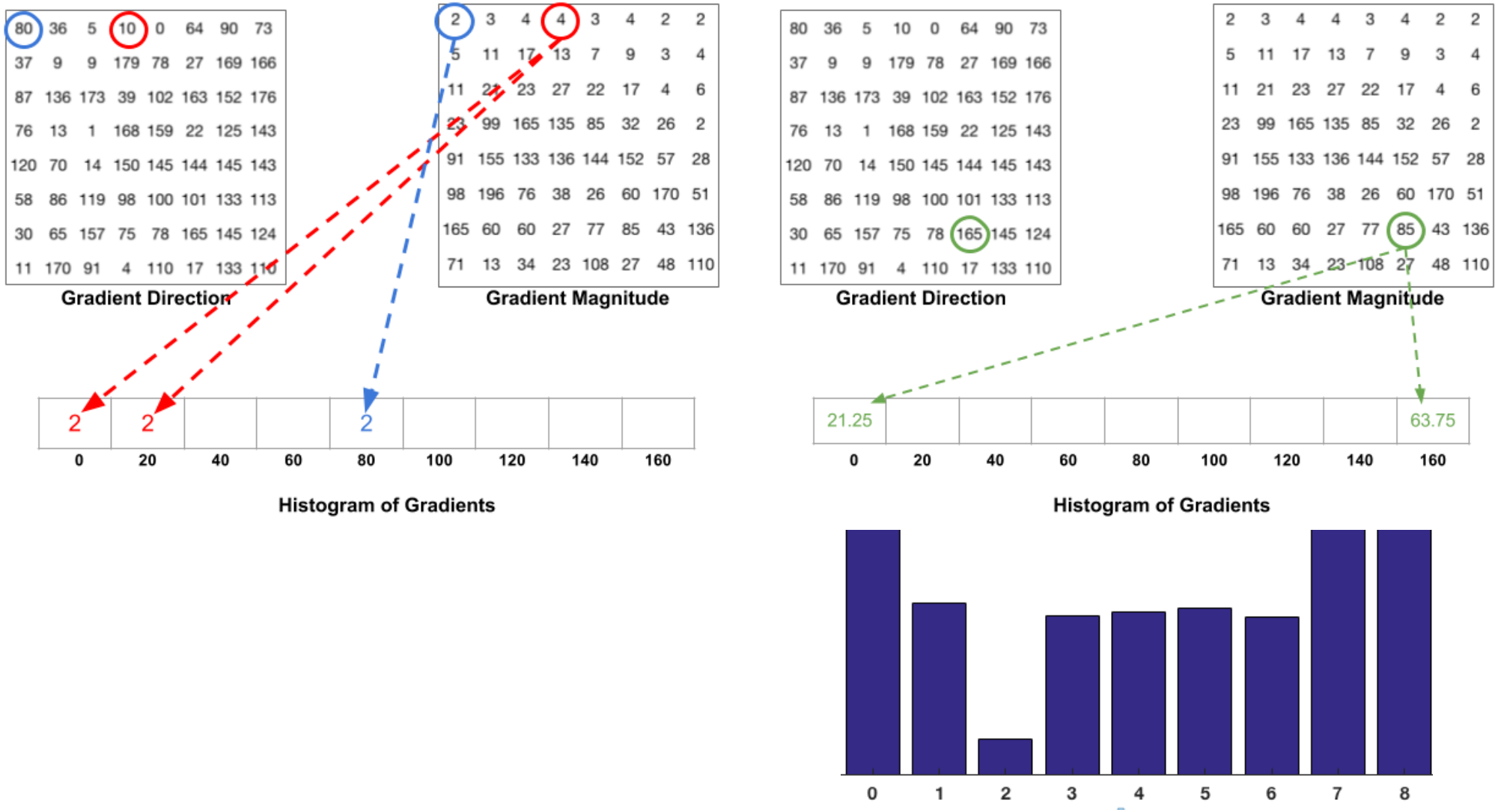
80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

가운데 : 화살표로 표시된 RGB 패치 및 Gradients
오른쪽 : 동일한 패치의 Gradients을 숫자로 표시

https://learnopencv.com/histogram-of-oriented-gradients/

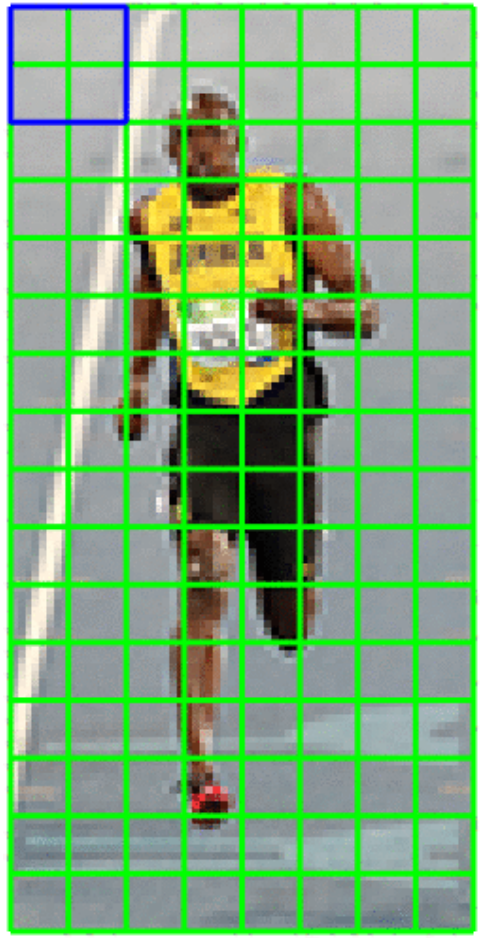
HOG – Histograms of Oriented Gradients



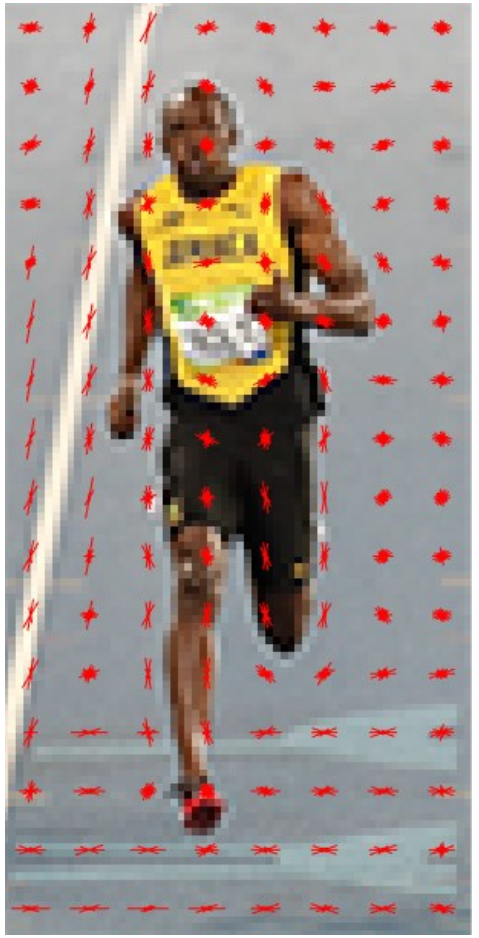
https://learnopencv.com/histogram-of-oriented-gradients/

HOG - Histograms of Oriented Gradients

Step 4 : 16×16 Block Normalization



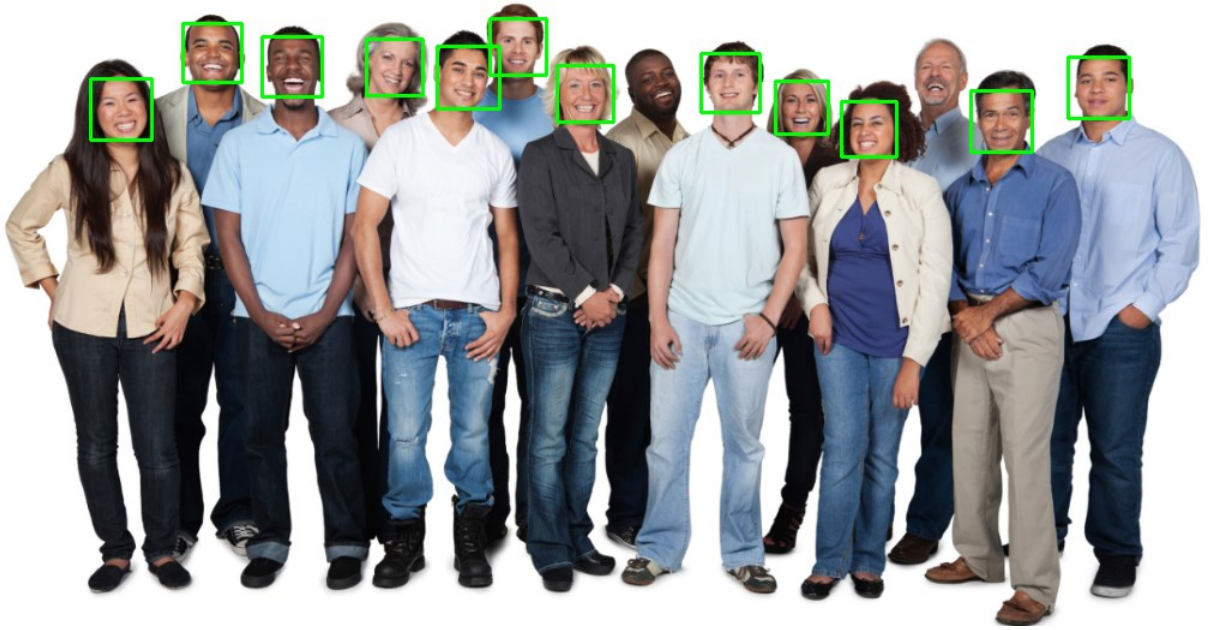
Step 5 : Calculate the Histogram of Oriented Gradients feature vector



https://learnopencv.com/histogram-of-oriented-gradients/

얼굴 탐지(Face Detection)

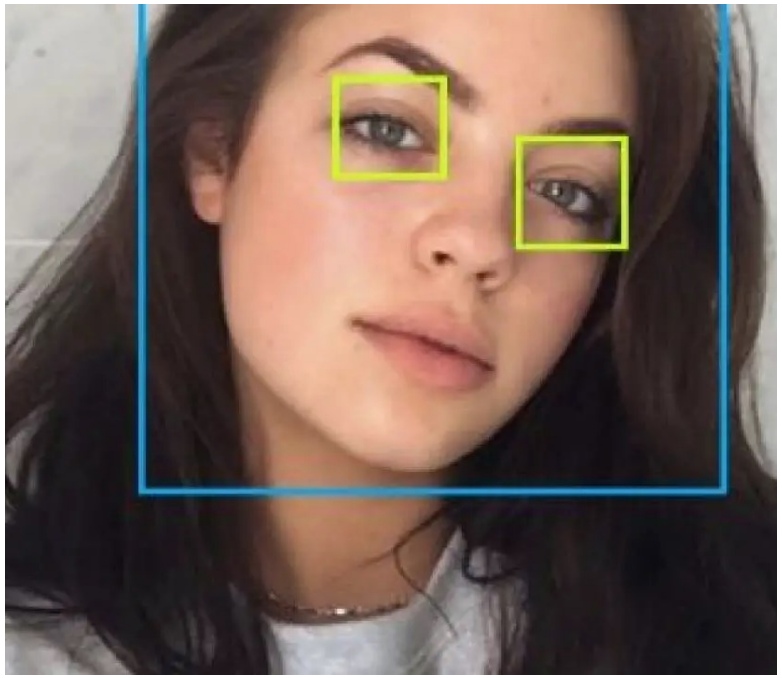
face_detection.ipynb



웹캠에서 얼굴 탐지

face_detector.py

```python
import cv2

face_detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eye_detector = cv2.CascadeClassifier("haarcascade_eye.xml")
cap = cv2.VideoCapture(0)

while True:
    # capture video frame
    ret, frame = cap.read()
    gray_image = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    detections = face_detector.detectMultiScale(
        gray_image, minSize=(100, 100), minNeighbors=5
    )

    # draw a rectangle around the faces
    for x, y, w, h in detections:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 255, 0), 2)
        rec_gray = gray_image[y : y + h, x : x + w]
        rec_color = frame[y : y + h, x : x + w]
        eyes = eye_detector.detectMultiScale(rec_gray)
        for x1, y1, w1, h1 in eyes:
            cv2.rectangle(rec_color, (x1, y1), (x1 + w1, y1 + h1), (0, 127, 255), 2)

    # display the resulting frame
    cv2.imshow("Face Recognition", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

# release the video capture
cap.release()
cv2.destroyAllWindows()
```



THANK YOU

kgpark88@gmail.com