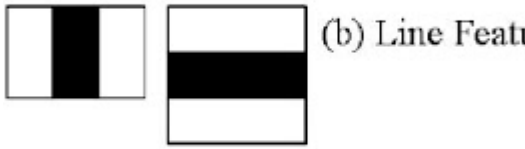
얼굴 탐지(Face Detection)

박경규

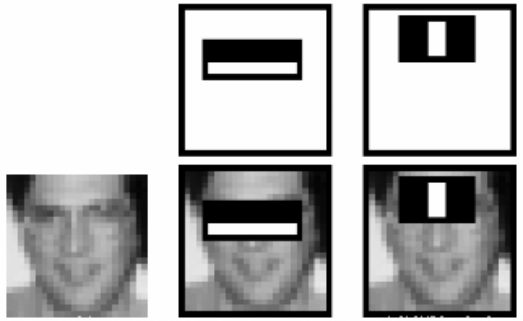
CASCADE CLASSIFIER

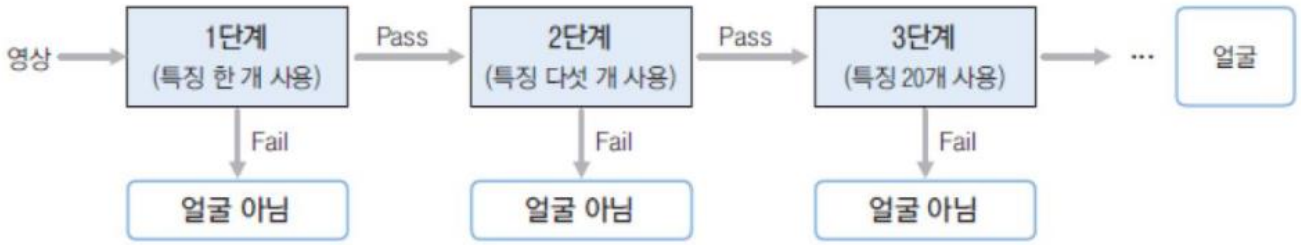
(a) Edge Features



(b) Line Features

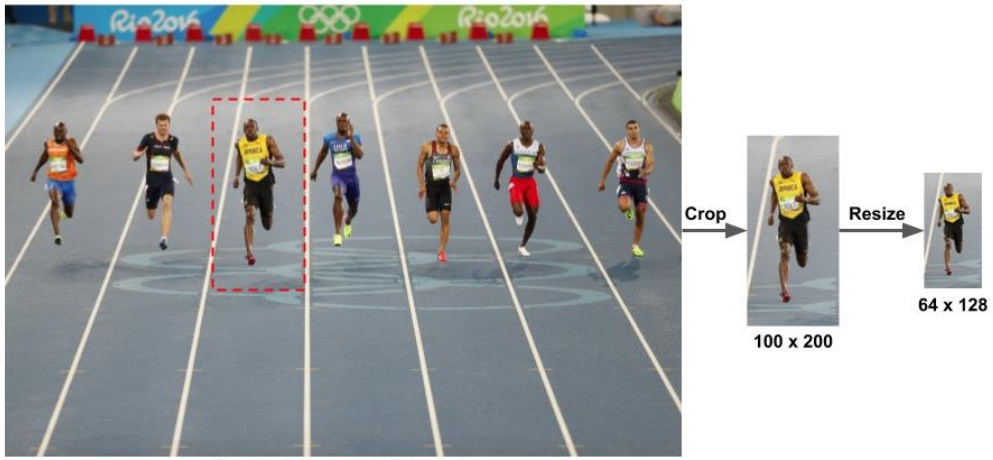
(c) Four-rectangle features



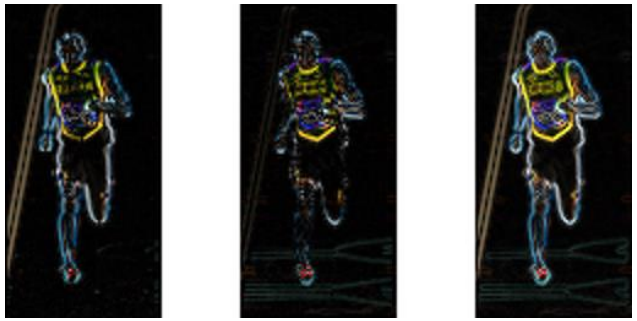


HOG - Histograms of Oriented Gradients

Step 1 : Preprocessing

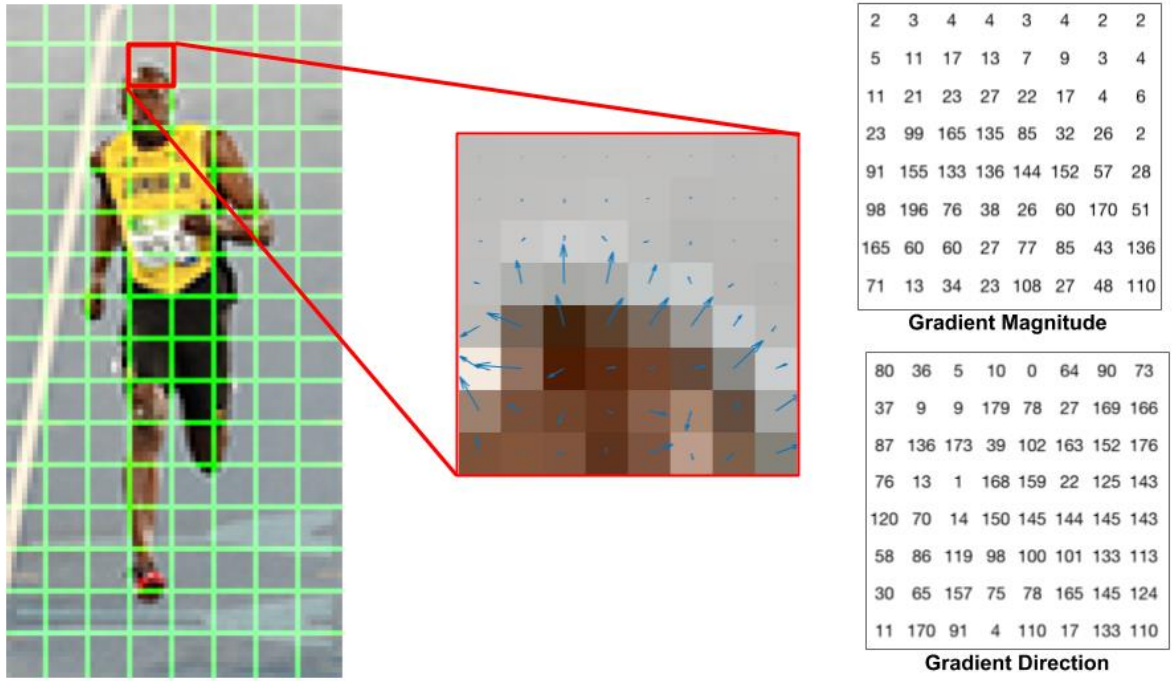


Step 2 : Calculate the Gradient Images



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

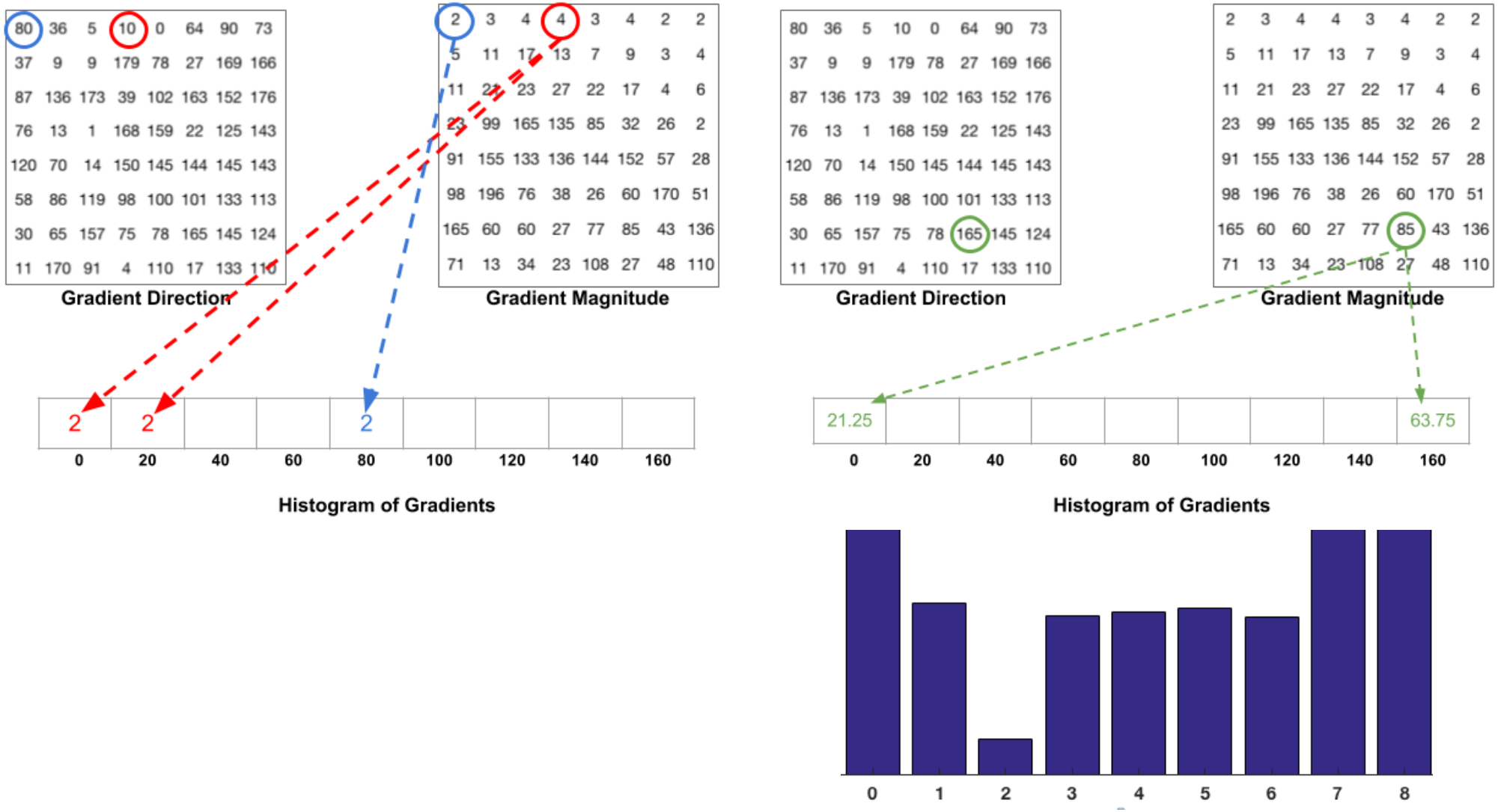
Step 3 : Calculate Histogram of Gradients in 8×8 cells



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

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

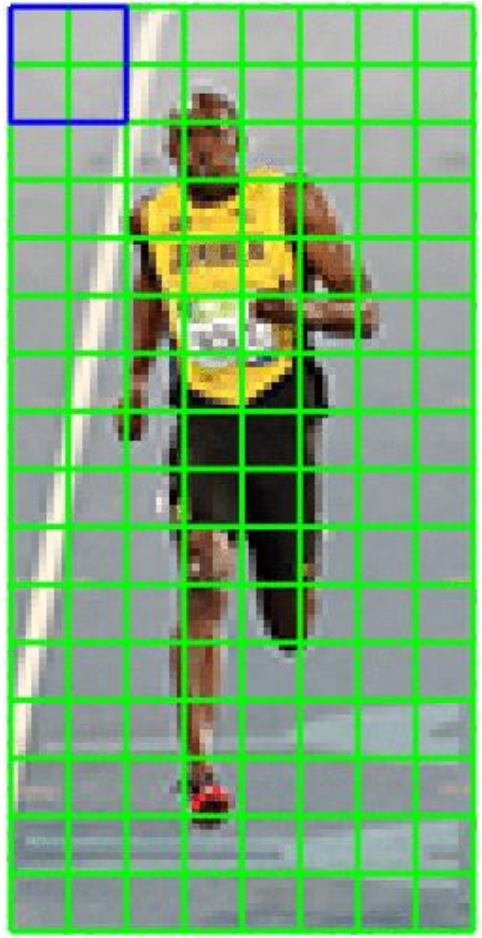
HOG – Histograms of Oriented Gradients



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

HOG - Histograms of Oriented Gradients

Step 4 : 16×16 Block Normalization

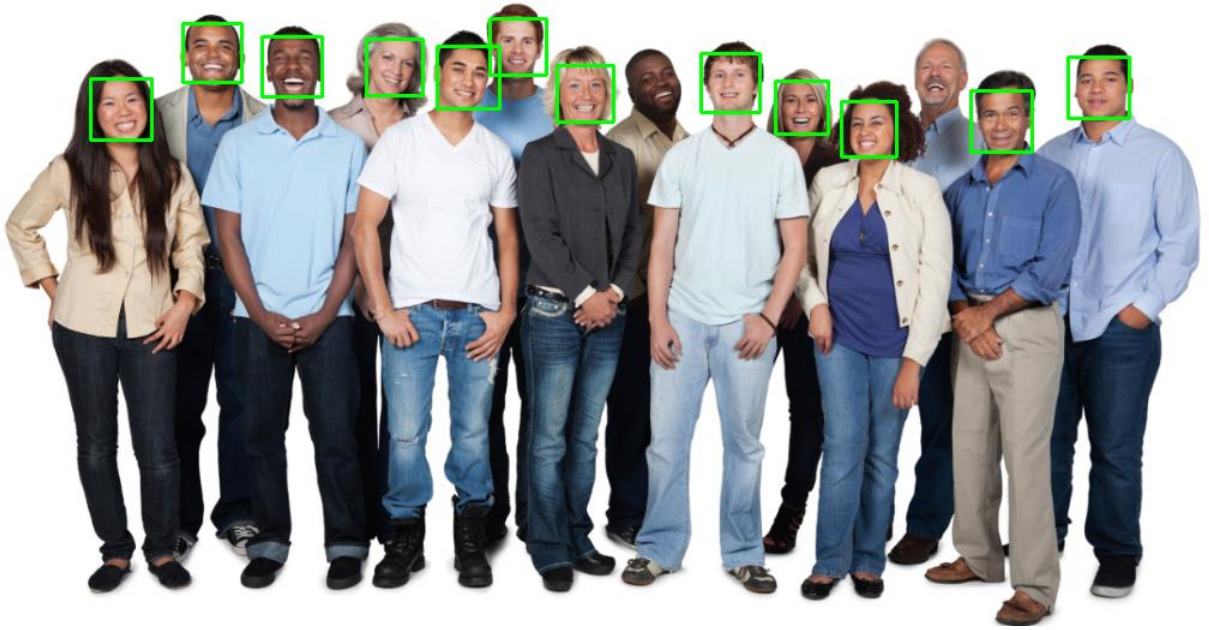


Step 5 : Calculate the Histogram of Oriented Gradients feature vector

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

얼굴 탐지(Face Detection)

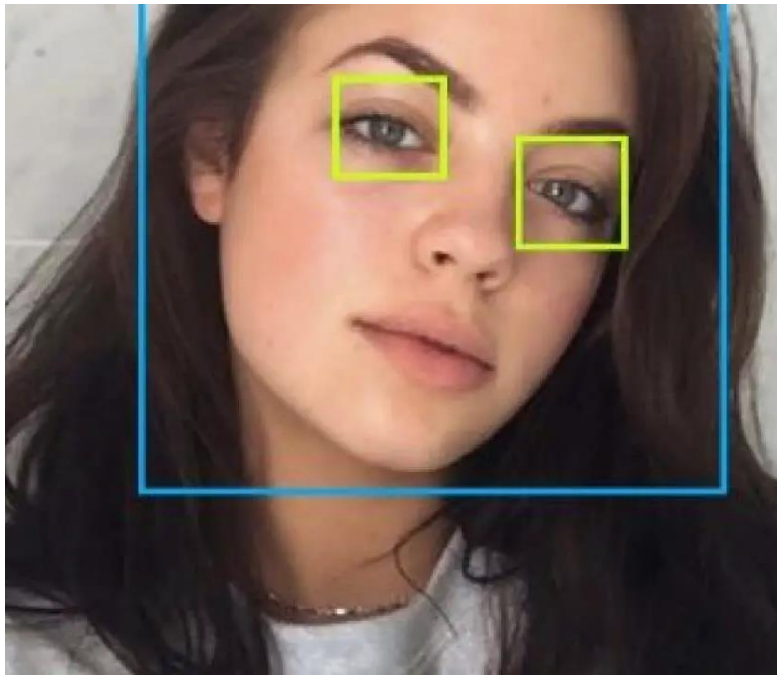
face_detection.ipynb



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