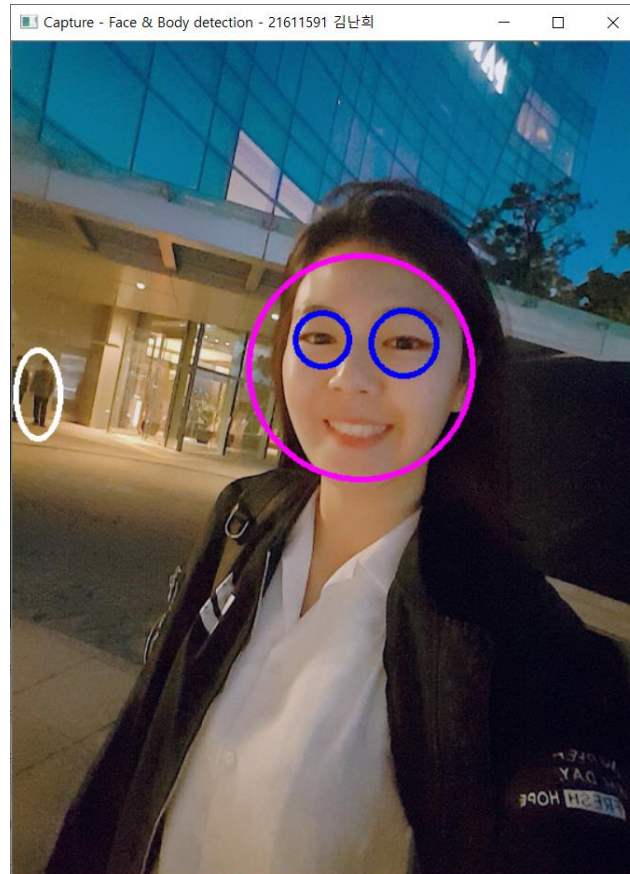


# [Computer Vision Programming]

## Lab 10. Face and Body detection

전자공학과 21611591 김난희

### -결과 사진



### -Discursion

full 샷이 찍혀 잘 서있는 내 사진이 따로 없어, 상반신만 있는 사진을 data로 사용하였다. 대신에 배경에 다른 사람의 body가 detection되는 사진을 뽑았다. 다른 full 샷을 이용해 테스트할 경우 body는 잘 detect하지만, 오히려 눈이 잘 detect 되지 않았다. 얼굴, 눈, 몸이 모두 detect 될 수 있도록 사진을 선정하였다.

### -소스 코드

```
#include <stdio.h>
#include <iostream>
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/objdetect/objdetect.hpp"

using namespace cv;
using namespace std;
```

```

/** Global variables */ //이곳에 support vector 등 으로 구성된 //미리 학습된 것을 가져와 사용함
String face_cascade_name = "C:/opencv24136/sources/data/haarcascades/haarcascade_frontalface_alt.xml";
String eyes_cascade_name = "C:/opencv24136/sources/data/haarcascades/haarcascade_eye_tree_eyeglasses.xml";
String fullbody_cascade_name = "C:/opencv24136/sources/data/haarcascades/haarcascade_fullbody.xml";

CascadeClassifier face_cascade; //얼굴 검출 클래스에서 객체 생성 -> 여기 안에 분류기 등 다 들어감
CascadeClassifier eyes_cascade; //눈 검출 클래스에서 객체 생성
CascadeClassifier fullbody_cascade; //full body 검출 클래스에서 객체 생성

string window_name = "Capture - Face & Body detection - 21611591 김난희";

void detectAndDisplay(Mat frame) /* @function detectAndDisplay */
{
    std::vector<Rect> faces;
    Mat frame_gray;
    cvtColor(frame, frame_gray, CV_BGR2GRAY); // color를 gray로 변환
    equalizeHist(frame_gray, frame_gray); // 대조비 키워줌 // EX. 밝은 건 1, 어두운 건 0

    //-- Detect faces
    face_cascade.detectMultiScale(frame_gray, faces, 1.1, 2, 0 | CV_HAAR_SCALE_IMAGE, Size(30, 30));
    //크기에 무관하게 detect하기 위함

    for (size_t i = 0; i < faces.size(); i++)
    {
        Point center(faces[i].x + faces[i].width*0.5, faces[i].y + faces[i].height*0.5);
        ellipse(frame, center, Size(faces[i].width*0.5, faces[i].height*0.5), 0, 0, 360,
        Scalar(255, 0, 255), 4, 8, 0); //얼굴을 찾아 분홍색의 원을 그림
        Mat faceROI = frame_gray(faces[i]); //얼굴 ROI

        std::vector<Rect> eyes;
        eyes_cascade.detectMultiScale(faceROI, eyes, 1.05, 2, 0 | CV_HAAR_SCALE_IMAGE, Size(30,
        30)); //In each face, detect eyes //얼굴에서 눈 검출
        for (size_t j = 0; j < eyes.size(); j++)
        {
            Point center(faces[i].x + eyes[j].x + eyes[j].width*0.5, faces[i].y + eyes[j].y +
            eyes[j].height*0.5);

            int radius = cvRound((eyes[j].width + eyes[j].height)*0.25);
            circle(frame, center, radius, Scalar(255, 0, 0), 4, 8, 0); // 찾은 눈에 파란색
            동그라미 그림
        }

        // full body도 같은 방법으로 찾음
        std::vector<Rect> bodys;
        fullbody_cascade.detectMultiScale(frame_gray, bodys, 1.1, 2, 0 | CV_HAAR_SCALE_IMAGE, Size(30,
        30)); //크기에 무관하게 detect하기 위함

        for (int j = 0; j < bodys.size(); j++)
        {
            Point center(bodys[j].x + bodys[j].width*0.5, bodys[j].y + bodys[j].height*0.5);
            ellipse(frame, center, Size(bodys[j].width*0.5, bodys[j].height*0.5), 0, 0, 360,
            Scalar(255, 255, 255), 4, 8, 0);
        }

        //-- Show what you got
        imshow(window_name, frame);
    }
}

int main(int argc, char** argv)
{
    //-- 1. Load the cascades DB (trained) //로드한 데이터 없으면 에러 띄움
    if (!face_cascade.load(face_cascade_name)) { printf("--(!)Error loading-face\n"); return -1; };
}

```

```
    if (!eyes_cascade.load(eyes_cascade_name)) { printf("--(!)Error loading-eyes\n"); return -1; };  
    if (!fullbody_cascade.load(fullbody_cascade_name)) { printf("--(!)Error loading-fullbody\n");  
return -1; };  
  
    //-- 2. Read the test image  
    Mat frame = imread("myimg2.jpg"); //cvQueryFrame( capture );  
    resize(frame, frame, Size(frame.size().width / 2, frame.size().height / 2));  
  
    //-- 3. Apply the classifier to the frame  
    detectAndDisplay( frame );  
    waitKey(0);  
    return 0;  
}
```