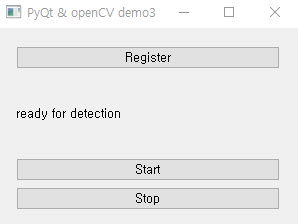
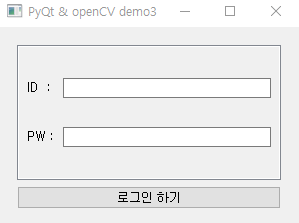
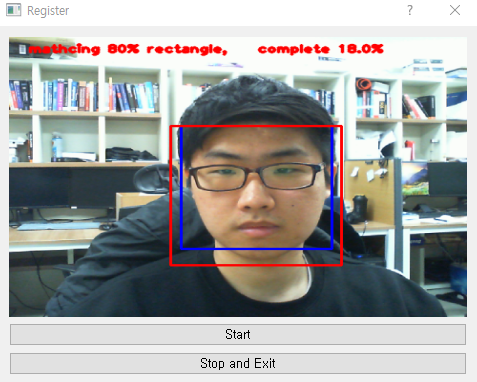
**1. face detection**

**- UI**

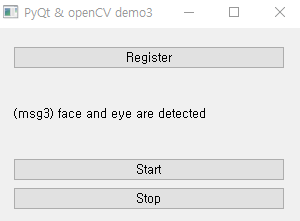
- 로그인 화면 & 기본 화면



- 등록 화면



- detection 화면



**- register**

ⓞ step0 : cam으로부터 얻을 수 있는 frame의 크기를 저장

① step1 : 30장의 얼굴을 먼저 인식하여, bounding box들의 평균으로 고정된 box 생성

이를 통해, 자리에 앉았을 때 얼굴이 위치하는 곳을 등록

(위의 UI에서 빨간색 box)

② step2 : step1에서 생성된 bounding box를 기준으로 현재의 얼굴을 찾아낸 box가

70%이상 면적이 겹칠 경우에만 사진을 수집(100장).

뿐만 아니라, ‘눈이 촬영된 횟수’와 ‘box의 좌표도 함께 저장’

③ step3 : 위로부터 얻은 정보들 저장(cam\_size.txt, eye\_count.txt, rectangle\_coord.txt)

수집된 150장의 사진도 저장

④ step4 : OpenCV recognizer를 이용하여 100장의 얼굴 학습한 후 저장(train.yml)

⑤ step5 : 학습된 recognizer로 새롭게 100장의 frame을 평가하여 confidence를 저장

**- detect**

① step1 : 5초마다 ‘촬영 이미지 횟수’, ‘얼굴 인식 횟수’, ‘눈 인식횟수’ 저장

5초동안의 confidence, rectangle\_coord 저장

**- 전송 :** 5초마다 새로 생기는 text 파일을 전송

① confidence.txt : 2~3KB

② count.txt : 1KB

③ rectangle\_coord.txt : 2~3KB

**- register index**

① cam\_size : cam으로 촬영된 frame의 크기. (480, 360) 고정

② eye\_count : 100장의 frame 중에서 눈의 인식된 횟수

③ confidence : 학습된 사진과 얼마나 다른지(error)를 나타내는 척도

Step5에서 새로운 100장의 frame으로 측정

④ rectangle\_coord : 등록 시 step1에서 생성된 bounding box의 좌표, 크기

**- detect index**

① img\_count : 5초 동안 찍힌 frame의 수

② face\_count : 5초 동안의 찍힌 frame 중에서 얼굴이 인식된 횟수

③ eye\_count : 5초 동안의 찍힌 frame 중에서 눈의 인식된 횟수

④ confidence : 학습 사진(train data 100장)과 얼마나 다른 지(error)를 나타내는 척도

⑤ rectangle\_coord : 매 frame에서 얻는 얼굴의 좌표

**- 파생변수**

① confidence의 min값, avg값 tracking

② 얼굴과 눈이 인식되는 비율

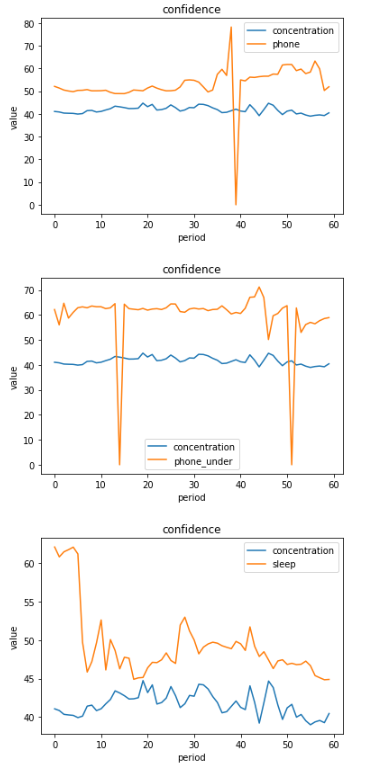
③ 중앙으로부터 rectangle\_coord가 얼마나 떨어져 있는지(위치 파악)

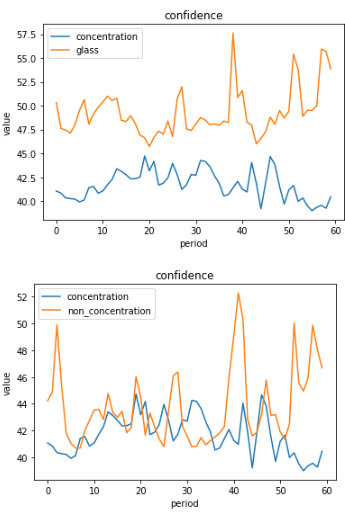
④ bounding box의 크기(화면에서 얼마나 떨어져 있는지)

⑤ registered rectangle과 겹치는 면적의 비율

⑥ 집중도 지표?

**- 실험**

****- confidence

****

1. 안경의 유무가 confidence의 차이를 만듦

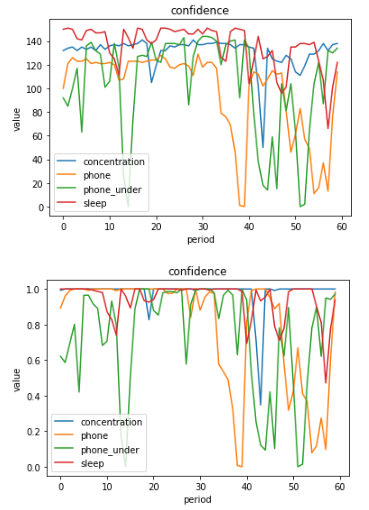
2. 핸드폰을 모니터 높이까지 들어서 한다면, 차이는 나지만, 구분선이 모호

3. 핸드폰을 책상에 내려놓고 하면, confidence의 차이가 커짐

4. 뒤로 기대서 잠을 자는 경우에도 confidence 차이가 발생

5. 컴퓨터로 다른 작업을 한다면, 집중하는지 안하는지 구별 불가

- face / face per img

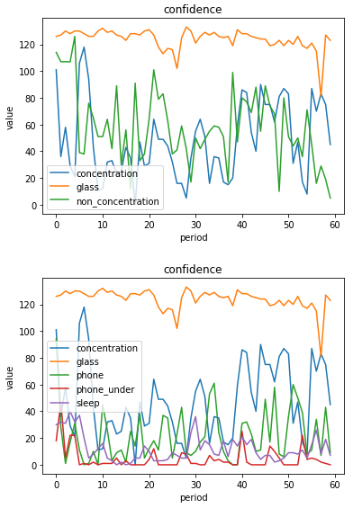


1. 얼굴인식은 잘 되는 편이라 그 숫자 자체로는 구별하기엔 무리가 있음

2. 하지만, 다른 행동을 하는 경우에는 얼굴인식을 못하는 경우가 많아서, 그래프의 형태가

아래로 많이 튀어나옴

- eye / eye per img

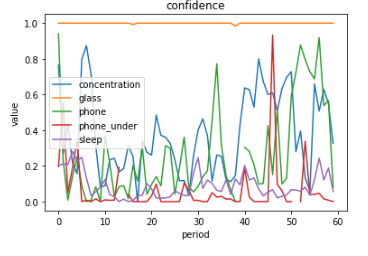
****

1. 안경을 쓰는 경우, 다른 case와 구분이 잘 안될 수도 있음

2. 이것은 안경을 쓸 경우 openCV가 완벽하게 눈을 구분하지 못하기 때문

3. 안경을 쓰지 않은 경우에는 유의미하게 구별됨

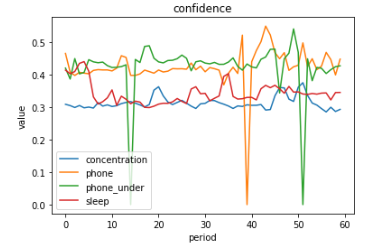
- eye per face



1. 숫자 자체로 구분하기엔 어렵지만, 다른 행동을 하는 경우 얼굴이 인식되어도 눈이 인식되지

않는 경우가 많아서 그래프가 위로 많이 튀어나옴

- concentration per img



**motion detection**

**- 목표** : 시험을 볼 때, 옆에서 친구가 있는지 확인

제대로 자리에 위치해 있는지 확인

움직임 등을 감지

엎드려 자는지 확인

**- 제약**

① 의자가 detection을 방해할 수도 있음

즉, 아래와 같이 의자가 나오지 않도록 옆에 cam을 설치해야 함

② 엎드려 자는 것을 확인하기 위해서는 책상이 보여야함

③ 모니터의 변화도 관찰되기 때문에 모니터도 보이면 안됨

④ 상대적 밝기에 매우 민감함

따라서, 창문/전구 등이 나와서는 안됨

⑤ 옆 사람도 가능하면 나오지 않아야 함

-> 너무 여러 개의 box가 잡혀서 정확히 탐지하는 것이 어려움

-> 옷색깔과 배경의 색과 겹칠 경우 인식을 할 수가 없음(검은색 옷 + 검은색 의자)

-> 카메라 각도 잡는 것이 거의 불가능

**- register**

ⓞ step0 : cam으로부터 얻을 수 있는 frame의 크기를 저장

① step1 : 고정된 자리를 촬영하여 저장(first frame)

**- detect**

① step1 : 저장해둔 사진(first frame)과 비교하여 얼마나 달라졌는지 확인

(엎드려 자는 경우에는 박스의 위치가 낮아지게 됨)

**- register index**

① cam\_size : cam으로 촬영된 frame의 크기. (640, 480) 고정

② rectangle\_coord : 등록 당시 bounding box의 좌표, 크기

(앉아 있는 자세, 엎드려 있는 자세)

**- detect index**

① rectangle\_coord : 등록 당시 bounding box의 좌표, 크기

**- index 이용 방법**

① rectangle\_coord의 정보로 '자리에 있는지', '엎드려 있는지'를 판별

**요약**

**- 전송하는 변수**

- 5초간 촬영된 이미지의 수

- 5초간 얼굴이 인식된 횟수

- 크기가 가장 큰 bounding box만 count

- 5초간 눈이 인식된 횟수

- confidence

- openCV 내부적으로 계산

- 얼굴이 인식되었을 경우에만 계산

- 5초 동안의 평균을 사용

- 얼굴이 인식되었을 때의 좌표

- x : 절대적인 x좌표

- y : 절대적인 y좌표

- w : bounding box의 가로길이

- h : bounding box의 세로길이

- 5초 동안의 평균을 사용

- (480, 360) 기준

- timestamp

- 위의 데이터들이 모두 계산되었을 때의 시간

- 5초 간격

**- 사용하는 변수**

- confidence

**주의**

**- 집중도와 참여도를 분리해서 생각하기**

**- 얼굴인식과 평가를 분리해서 생각하기**

**- 환경이 달라져도 인식할 수 있어야함**

**- 일단은 사람이 바뀐다는 생각은 하지 않고 진행**

**- confidence가 높아질 경우에 재학습**

**- 인텔리빅스와 비교하기**

**- 학습기간의 평균과 분산을 기준으로 어느정도 수준을 벗어나면 이상하다고 판단**