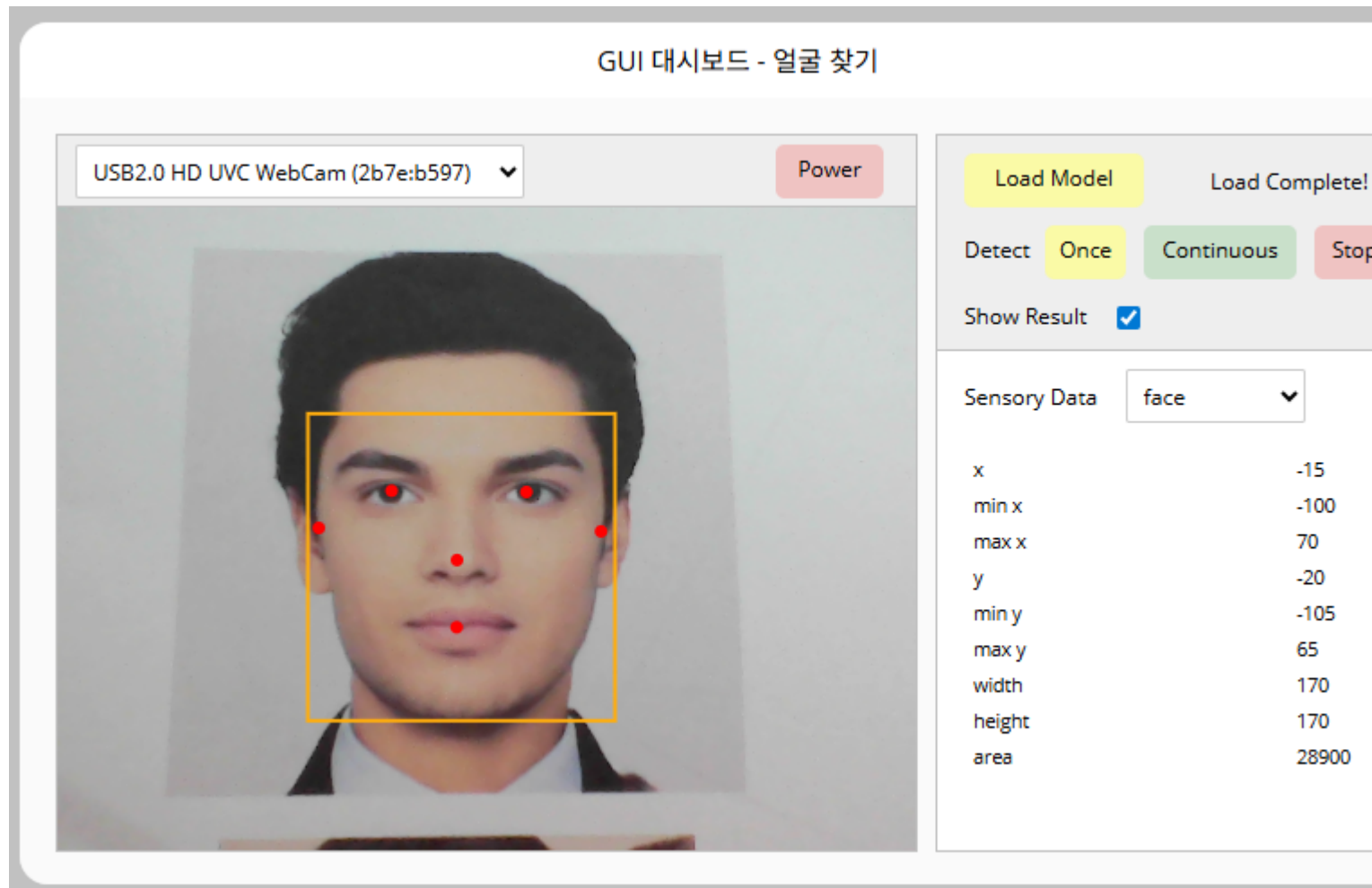


대시보드 열기

대시보드 열기 는 블록 코딩에 사용할 수 있는 블록은 아니지만, 확장 모듈에서 사용되는 모델이 어떠한 방식으로 적용되는지 확인하는 대시보드를 열 수 있습니다.

대시보드 화면

대시보드 열기 클릭 시 다음과 같은 화면을 볼 수 있습니다.



세부 버튼

Power

선택한 카메라를 키거나 끕니다.

Load Model

학습된 얼굴 모델을 불러옵니다. ‘얼굴 찾기’확장 모듈을 사용하기 위해서 반드시 필요한 작업입니다.

Detect

얼굴 찾기를 실행하거나 멈춥니다.

Once 버튼으로 한번만 실행할 지, Continuous 버튼으로 연속으로 실행할 지 정할 수 있습니다.

또한, Stop 버튼을 통해 찾기를 멈출 수 있습니다.

Show Result

얼굴 찾기 결과를 카메라 화면 상으로 출력합니다.

Sensory Data

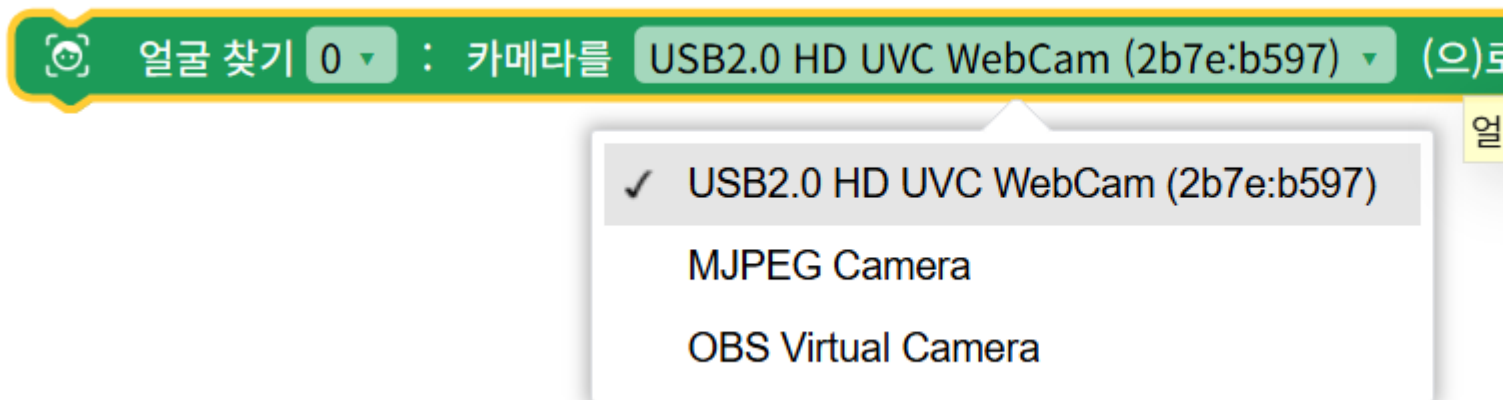
얼굴 찾기에서 찾은 얼굴 데이터 값을 출력합니다.

선택한 얼굴 부위의 x,y 좌표를 확인할 수 있습니다. ##### 드롭다운 옵션 및 입력값 | 이름 | 구분 | 설명 | 범위 / 종류 |
| — | — | — | — | | part | 드롭다운 옵션 | 얼굴 부위 | face, ear, eye, nose, mouth |

블록

카메라 정하기

얼굴 찾기 모듈에 사용할 카메라를 선택합니다.



드롭다운 옵션 및 입력값

이름	구분	설명	범위 / 종류
camera	드롭다운 옵션	사용할 카메라	연결한 카메라 리스트

자바스크립트 코드

```
// 특정 카메라를 얼굴 찾기를 위한 카메라로 정하기 (id 는 예시)
$('FaceDetection*0:camera.deviceId').d =
'035658da47183882a695a82c45b8f3e9ae50cef47945ccdc3f31e1ae1fbca9cb';
```

파이썬 코드

```
# 특정 카메라를 얼굴 찾기를 위한 카메라로 정하기 (id 는 예시)
__('FaceDetection*0:camera.deviceId').d =
'035658da47183882a695a82c45b8f3e9ae50cef47945ccdc3f31e1ae1fbca9cb'
```

얼굴 모델 불러오기

학습된 얼굴 모델을 불러옵니다. '얼굴 찾기' 모듈의 기능들을 사용하기 위해서는 이 작업이 반드시 필요합니다.

기다리기를 체크하면, 모델 불러오기가 완료될 때까지 기다립니다.

단, 기다리기를 체크한 경우에는 async 함수 내에서만 사용할 수 있습니다.



학습된 얼굴 모델을 불러옵니다. '얼굴 찾기' 모듈의 기능들을 사용하기

자바스크립트 코드

```
// 얼굴 모델 불러오기 | 기다리기 0
$('FaceDetection*0:load_model').d = 1;
await $('FaceDetection*0:!load_model').w();

// 얼굴 모델 불러오기 | 기다리기 X
$('FaceDetection*0:load_model').d = 1;
```

파이썬 코드

```
# 얼굴 모델 불러오기 | 기다리기 0
__('FaceDetection*0:load_model').d = 1
await __('FaceDetection*0:!load_model').w()

# 얼굴 모델 불러오기 | 기다리기 X
__('FaceDetection*0:load_model').d = 1
```

얼굴 한 번 찾기

현재 화면에 있는 얼굴을 찾아 딱 한번 표시합니다.



현재 화면에 있는 얼굴을 찾아 딱 한번 표시합니다.

자바스크립트 코드

```
// 얼굴 한 번 찾기
$('FaceDetection*0:detect.once').d = 1;
```

파이썬 코드

```
# 얼굴 한 번 찾기
__('FaceDetection*0:detect.once').d = 1
```

얼굴 연속으로 찾기

얼굴 연속으로 찾기를 시작하거나 중지합니다.

얼굴 연속으로 찾기를 시작하면, 현재 화면에 있는 얼굴을 계속 따라가며 화면상에 표시합니다.


얼굴 찾기 0 : 연속으로 얼굴 찾기
시작하기

현재 화면에 있는 얼굴을 계속

✓ 시작하기

중지하기

드롭다운 옵션 및 입력값

이름	구분	설명	범위 / 종류
toggle	드롭다운 옵션	얼굴 찾기	시작하기 (1), 중지하기 (0)

자바스크립트 코드

```
// 연속으로 얼굴 찾기 시작하기
$('FaceDetection*0:detect.continuous').d = 1;

// 연속으로 얼굴 찾기 중지하기
$('FaceDetection*0:detect.continuous').d = 0;
```

파이썬 코드

```
# 연속으로 얼굴 찾기 시작하기
__('FaceDetection*0:detect.continuous').d = 1

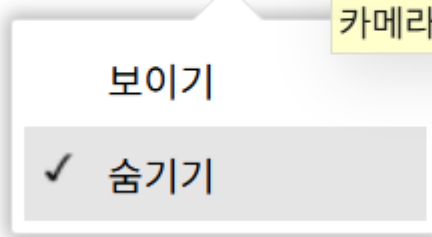
# 연속으로 얼굴 찾기 중지하기
__('FaceDetection*0:detect.continuous').d = 0
```

얼굴 찾기 결과 보이기

카메라 화면에 얼굴 찾기 결과를 표시할지 말지를 결정합니다.



카메라 화면에 얼굴 찾기 결과를 표시할지 말지를



드롭다운 옵션 및 입력값

이름	구분	설명	범위 / 종류
toggle	드롭다운 옵션	얼굴 찾기 결과	보이기 (1), 숨기기 (0)

자바스크립트 코드

```
// 얼굴 찾기 결과 보이기
$('#FaceDetection*0:display').d = 1;

// 얼굴 찾기 결과 숨기기
$('#FaceDetection*0:display').d = 0;
```

파이썬 코드

```
# 얼굴 찾기 결과 보이기
__('FaceDetection*0:display').d = 1

# 얼굴 찾기 결과 숨기기
__('FaceDetection*0:display').d = 0
```

얼굴 데이터

얼굴 찾기로 얻은 얼굴 데이터를 반환합니다.


 얼굴 찾기 0 : 얼굴 x 좌표

- ✓ 얼굴
- 왼쪽 눈
- 오른쪽 눈
- 왼쪽 귀
- 오른쪽 귀
- 코
- 입

선택한 얼굴 데이터

이름	구분	설명	범위 / 종류
part	드롭다운 옵션	얼굴 부위	얼굴 (face), 왼쪽 눈 (eye.left), 오른쪽 눈 (eye.right), 왼쪽 귀 (ear.left), 오른쪽 귀 (ear.right), 코 (nose), 입 (mouth)
axis	드롭다운 옵션	좌표	x, y

자바스크립트 코드

```

// 얼굴의 x 좌표
$('#FaceDetection*0:face.x').d;

// 얼굴의 y 좌표
$('#FaceDetection*0:face.y').d;

```

```
// 왼쪽 눈의 x 좌표
$('#FaceDetection*0:eye.left.x').d;

// 왼쪽 눈의 y 좌표
$('#FaceDetection*0:eye.left.y').d;

// 오른쪽 눈의 x 좌표
$('#FaceDetection*0:eye.right.x').d;

// 오른쪽 눈의 y 좌표
$('#FaceDetection*0:eye.right.y').d;

// 왼쪽 귀의 x 좌표
$('#FaceDetection*0:ear.left.x').d;

// 왼쪽 귀의 y 좌표
$('#FaceDetection*0:ear.left.y').d;

// 오른쪽 귀의 x 좌표
$('#FaceDetection*0:ear.right.x').d;

// 오른쪽 귀의 y 좌표
$('#FaceDetection*0:ear.right.y').d;

// 코의 x 좌표
$('#FaceDetection*0:nose.x').d;

// 코의 y 좌표
$('#FaceDetection*0:nose.y').d;

// 입의 x 좌표
$('#FaceDetection*0:mouth.x').d;

// 입의 y 좌표
```



```
$('#FaceDetection*0:mouth.y').d;
```

파이썬 코드

```
# 얼굴의 x 좌표
__('FaceDetection*0:face.x').d

# 얼굴의 y 좌표
__('FaceDetection*0:face.y').d

# 왼쪽 눈의 x 좌표
__('FaceDetection*0:eye.left.x').d

# 왼쪽 눈의 y 좌표
__('FaceDetection*0:eye.left.y').d

# 오른쪽 눈의 x 좌표
__('FaceDetection*0:eye.right.x').d

# 오른쪽 눈의 y 좌표
__('FaceDetection*0:eye.right.y').d

# 왼쪽 귀의 x 좌표
__('FaceDetection*0:ear.left.x').d

# 왼쪽 귀의 y 좌표
__('FaceDetection*0:ear.left.y').d

# 오른쪽 귀의 x 좌표
__('FaceDetection*0:ear.right.x').d

# 오른쪽 귀의 y 좌표
__('FaceDetection*0:ear.right.y').d
```

```
# 코의 x 좌표
__('FaceDetection*0:nose.x').d

# 코의 y 좌표
__('FaceDetection*0:nose.y').d

# 입의 x 좌표
__('FaceDetection*0:mouth.x').d

# 입의 y 좌표
__('FaceDetection*0:mouth.y').d
```

얼굴 주변 사각형 데이터

얼굴 찾기로 찾은 얼굴의 주변을 사각형으로 정의하여, 그 사각형의 데이터를 반환합니다.



얼굴 주변 사각형에 대한 데이터

✓ x 좌표 최솟값

x 좌표 최댓값

y 좌표 최솟값

y 좌표 최댓값

폭

높이

넓이

이름	구분	설명	범위 / 종류
data	드롭다운 옵션	얼굴 데이터	x 좌표 최솟값 (min_x), x 좌표 최댓값 (max_x), y 좌표 최솟값 (min_y), y 좌표 최댓값 (max_y), 폭 (width), 높이 (height), 넓이 (area)

자바스크립트 코드

```
// 얼굴 사각형 x 좌표 최솟값
$('FaceDetection*0:face.min_x').d;

// 얼굴 사각형 x 좌표 최댓값
$('FaceDetection*0:face.max_x').d;

// 얼굴 사각형 y 좌표 최솟값
$('FaceDetection*0:face.min_y').d;

// 얼굴 사각형 y 좌표 최댓값
$('FaceDetection*0:face.max_y').d;

// 얼굴 사각형 폭
$('FaceDetection*0:face.width').d;

// 얼굴 사각형 높이
$('FaceDetection*0:face.height').d;

// 얼굴 사각형 넓이
$('FaceDetection*0:face.area').d;
```

파이썬 코드

```
# 얼굴 사각형 x 좌표 최솟값
__('FaceDetection*0:face.min_x').d

# 얼굴 사각형 x 좌표 최댓값
__('FaceDetection*0:face.max_x').d

# 얼굴 사각형 y 좌표 최솟값
__('FaceDetection*0:face.min_y').d

# 얼굴 사각형 y 좌표 최댓값
__('FaceDetection*0:face.max_y').d

# 얼굴 사각형 폭
__('FaceDetection*0:face.width').d

# 얼굴 사각형 높이
__('FaceDetection*0:face.height').d

# 얼굴 사각형 넓이
__('FaceDetection*0:face.area').d
```

얼굴 요소 사이 거리

얼굴 찾기로 찾은 얼굴의 데이터를 이용하여 두 얼굴 요소 사이의 거리를 반환합니다.

얼굴 찾기 0 : 왼쪽 눈 에서 왼쪽 눈 까지 거리

두 얼굴 요소 사

✓ 거리

가로 거리

세로 거리

이름	구분	설명	범위 / 종류
part1	드롭다운 옵션	얼굴 부위	얼굴 (face), 왼쪽 눈 (eye.left), 오른쪽 눈 (eye.right), 왼쪽 귀 (ear.left), 오른쪽 귀 (ear.right), 코 (nose), 입 (mouth)
part2	드롭다운 옵션	얼굴 부위	얼굴 (face), 왼쪽 눈 (eye.left), 오른쪽 눈 (eye.right), 왼쪽 귀 (ear.left), 오른쪽 귀 (ear.right), 코 (nose), 입 (mouth)
distance	드롭다운 옵션	거리	거리 (distance), 가로거리 (horizontal distance), 세로거리 (vertical distance)

자바스크립트 코드

```
// 왼쪽 눈에서 왼쪽 눈 까지 거리
Math.sqrt( Math.pow(($('FaceDetection*0:eye.left.x').d - $('FaceDetection*0:eye.left.x').d), 2)
+ Math.pow(($('FaceDetection*0:eye.left.y').d - $('FaceDetection*0:eye.left.y').d), 2) );

// 왼쪽 눈에서 오른쪽 눈 까지 가로거리
```

```

Math.abs($('FaceDetection*0:eye.right.x').d - $('FaceDetection*0:eye.left.x').d);

// 왼쪽 눈에서 왼쪽 귀까지 세로거리
Math.abs($('FaceDetection*0:ear.left.y').d - $('FaceDetection*0:eye.left.y').d);

// 왼쪽 눈에서 오른쪽 귀까지 거리
Math.sqrt( Math.pow(($('FaceDetection*0:ear.right.x').d - $('FaceDetection*0:eye.left.x').d), 2)
+ Math.pow(($('FaceDetection*0:ear.right.y').d - $('FaceDetection*0:eye.left.y').d), 2) );

// 왼쪽 눈에서 코까지 거리
Math.sqrt( Math.pow(($('FaceDetection*0:nose.x').d - $('FaceDetection*0:eye.left.x').d), 2) +
Math.pow(($('FaceDetection*0:nose.y').d - $('FaceDetection*0:eye.left.y').d), 2) );

// 왼쪽 눈에서 입까지 거리
Math.sqrt( Math.pow(($('FaceDetection*0:mouth.x').d - $('FaceDetection*0:eye.left.x').d), 2) +
Math.pow(($('FaceDetection*0:mouth.y').d - $('FaceDetection*0:eye.left.y').d), 2) );

```

파이썬 코드

```

# 왼쪽 눈에서 왼쪽 눈 까지 거리
math.sqrt( math.pow((__('FaceDetection*0:eye.left.x').d - __('FaceDetection*0:eye.left.x').d),
2) + math.pow((__('FaceDetection*0:eye.left.y').d - __('FaceDetection*0:eye.left.y').d), 2) )

# 왼쪽 눈에서 오른쪽 눈 까지 가로거리
math.fabs((__('FaceDetection*0:eye.right.x').d - __('FaceDetection*0:eye.left.x').d)

# 왼쪽 눈에서 왼쪽 귀까지 세로거리
math.fabs((__('FaceDetection*0:ear.left.y').d - __('FaceDetection*0:eye.left.y').d)

# 왼쪽 눈에서 오른쪽 귀까지 거리
math.sqrt( math.pow((__('FaceDetection*0:ear.right.x').d - __('FaceDetection*0:eye.left.x').d),
2) + math.pow((__('FaceDetection*0:ear.right.y').d - __('FaceDetection*0:eye.left.y').d), 2) )

```

왼쪽 눈에서 코까지 거리

```
math.sqrt( math.pow((__('FaceDetection*0:nose.x').d - __('FaceDetection*0:eye.left.x').d), 2) +  
math.pow((__('FaceDetection*0:nose.y').d - __('FaceDetection*0:eye.left.y').d), 2) )
```

왼쪽 눈에서 입까지 거리

```
math.sqrt( math.pow((__('FaceDetection*0:mouth.x').d - __('FaceDetection*0:eye.left.x').d), 2) +  
math.pow((__('FaceDetection*0:mouth.y').d - __('FaceDetection*0:eye.left.y').d), 2) )
```

얼굴 모델 로딩 상태값

얼굴 모델 로딩 상태를 반환합니다.

아직 불러오지 않았다면 0, 불러오는 중이면 1, 불러오기를 완료했다면 2 를 반환합니다.



얼굴 모델 로딩 상태를 반환합니다.

아직 불러오지 않았으면 0, 불러오는 중이면 1, 불러오기를 완료했

자바스크립트 코드

```
// 얼굴 모델 로딩 상태 값  
$('#FaceDetection*0:model_state').d;
```

파이썬 코드

```
# 얼굴 모델 로딩 상태 값  
__('FaceDetection*0:model_state').d
```

얼굴을 찾았는가?

얼굴 찾기 여부를 참 (1) / 거짓 (0) (으) 로 반환합니다.

 얼굴 찾기 0 ▼ : 얼굴을 찾았는가?

얼굴을 찾았는지 여부

자바스크립트 코드

```
// 얼굴을 찾았는가?  
$('FaceDetection*0:detected').d;
```

파이썬 코드

```
# 얼굴을 찾았는가?  
__('FaceDetection*0:detected').d
```