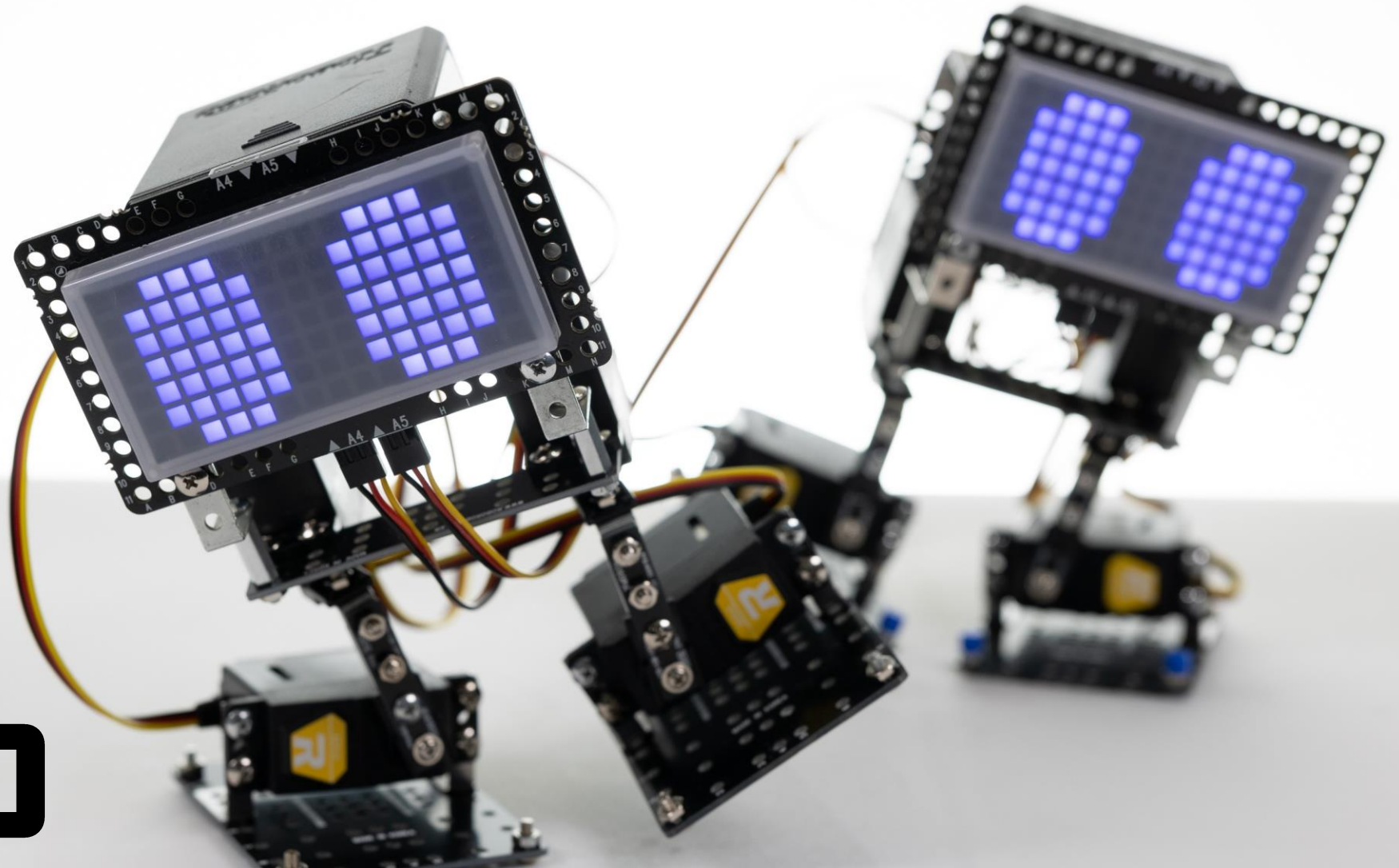


AI 이족보행 로봇 [Walker]

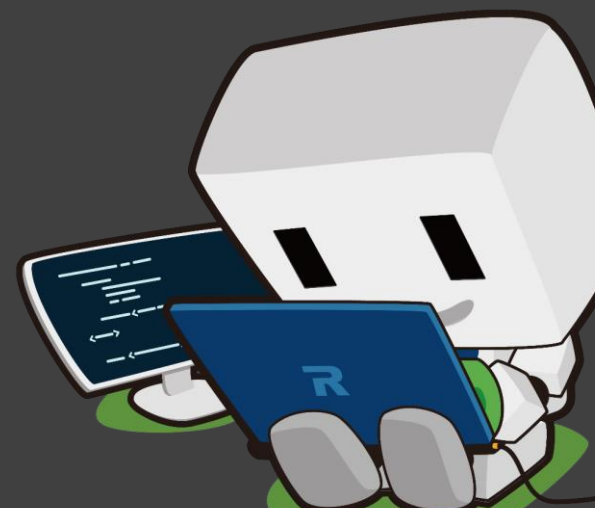
Hi



DIYGO

Step. 11 Python AI Coding

- 영상인식 기능학습
- 영상인식을 통해 로봇제어하기





Python으로 [워크] 제어하기

Visual Studio Code에서 인공지능 기능 활용하기

해당 차시는..

Python 구문을 활용해서, 얼굴인식, ArUco 마커를 인식할 수 있는 간단한 체험형 실습을 진행하도록 구성되어 있습니다. (Python문법 등에 대한 보강자료 없음)

Python과 Visual Studio Code, 영상인식 라이브러리를 활용한 실습을 진행할 예정이며, 기본 명령어를 따라서 타이핑 해 보며 노트북의 웹캠 또는 PC에 연결된 웹캠을 이용해 인공지능 실습을 진행해 봅니다.

Python 및 Visual Studio Code의 업데이트 상황에 따라 표기되는 명칭 및 이미지가 다를 수 있습니다.



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩 준비하기



노트북 웹캠

또는

PC와 연결된 유선 웹캠이 필요해요.



Python으로 [워크] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - 기본 세팅 (라이브러리 설치하기)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\05_자습\자습\Python test> pip3 install Robocam
```

영상인식 라이브러리 설치 명령어

터미널에 사진과 같이 영상인식 라이브러리 설치 명령어를 입력합니다.

```
pip3 install RoboCam
```



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - 기본 세팅

```
Python test.py > ...  
1  from RoboCam.robocam import *  
2  rCam = RoboCam()  
3  rCam.WebcamStreamInit()  
4  rCam.WebcamStream()  
5  input()
```

RoboCam 라이브러리 중 robocam의 항목을 호출

RoboCam 클래스로

'rCam'이라는 이름을 가진 객체 생성

input() 명령어를 통해 외부입력을 받으면 프로그램 정지

WebcamStreamInit (width : Int = 640, height : int = 480)

- 웹캠의 스트리밍을 준비하는 함수예요. width * height의 해상도로 스트리밍 됩니다.

* Width -> 가로 해상도 (기본값 : 640)

* height -> 세로 해상도 (기본값 : 480)

WebcamStream()

- 웹캠의 영상 스트리밍을 시작하는 함수예요.

* WebcamStreamInit 함수가 호출된 후 사용되어야 해요.



Python으로 [워커] 제어하기

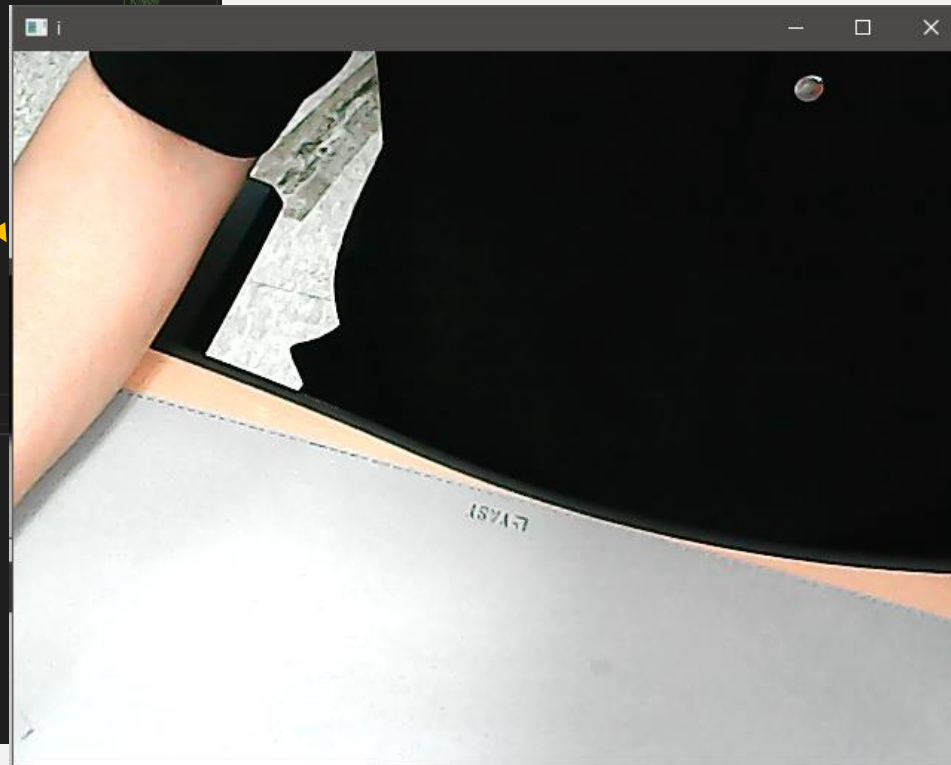
Visual Studio Code를 활용한 Python 인공지능 실습

스트리밍화면 확인하기

```
Python test.py > ...  
1 from RoboCam.robocam import *  
2 rCam = RoboCam()  
3 rCam.WebcamStreamInit()  
4 rCam.WebcamStream()  
5 input()  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16
```

실행시 스트리밍 화면 생성

터미널 창에 '엔터'키 입력시 스트리밍 종료





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - Python 명령어 알기

스트리밍된 화면 속성을 제어하는 명령어 함수들에 대해 알아보까요?

LeftRightFlipMode(flag:bool)

- 웹캠의 영상 스트리밍을 flag에 따라 좌우 반전시키는 함수입니다.
- * flag -> True: 좌우반전 On , False : 좌우반전 Off

UpDownFlipMode(flag:bool)

- 웹캠의 영상 스트리밍을 flag에 따라 상하 반전시키는 함수입니다.
- * flag -> True: 상하반전 On , False : 상하반전 Off

MosaicMode(rate:int)

- 웹캠의 영상 스트리밍에 모자이크를 rate비율로 설정하는 함수입니다.
- * rate -> 영상 모자이크의 비율입니다. [단위 : 퍼센트(%), 기본값:0]



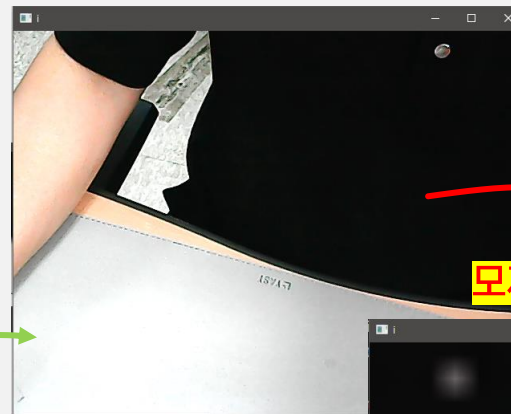
Python으로 [워크] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - Python 명령어 실습

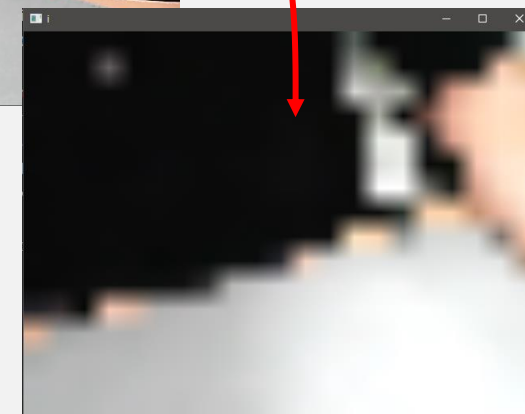
학습한 명령어를 이용해, 아래처럼 화면의 속성을 변경할 수 있어요.

```
1 from RoboCam.robocam import *
2 rCam = RoboCam()
3 rCam.WebcamStreamInit()
4 rCam.WebcamStream()
5 rCam.LeftRightFlipMode(True)
6 rCam.MosaicMode(30)
7 input()
```



좌우반전

모자이크



화면반전과 모자이크 처리를 통해 우측의 그림처럼 스트리밍 되는 화면의 속성을 변경했어요.

모자이크 비율 / 반전방향 등을 변경해서 코드를 작성해 보세요.



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - Python 명령어 알기

ArUco 마커의 상세 데이터를 확인하는 명령어 함수에 대해 알아보까요?

`ArucoDetectorInit()`

- Aruco마커 탐색 모듈을 초기화하는 함수입니다.
(Original ArUco 마커를 타겟합니다)

`ArucoDetectorStart()`

- Aruco마커 탐색 모듈을 실행 시키는 함수입니다.
* 탐색 모듈을 중지하는 함수는 `ArUcoDetectorStop()` 입니다.

`DrawArucoArea(flag:bool)`

- 화면에 인식된 Aruco마커의 영역표시 여부를 정하는 함수입니다.
* 아래부터 나열될 상세데이터 함수는 '영역표시'가 활성화 될 때 사용할 수 있어요.

`DrawArucoId(flag:bool)`

- 화면에 인식된 Aruco마커의 ID표시 여부를 정하는 함수입니다.



Python으로 [위커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - Python 명령어 알기

ArUco 마커의 상세 데이터를 확인하는 명령어 함수에 대해 알아보까요?

`DrawArucoPoint(flag:bool)`

- 화면에 인식된 Aruco마커의 좌표표시 여부를 정하는 함수입니다.

`DrawArucoSize(flag:bool)`

- 화면에 인식된 Aruco마커의 크기표시 여부를 정하는 함수입니다.

`DrawArucoDistance(flag:bool)`

- 화면에 인식된 Aruco마커의 거리표시 여부를 정하는 함수입니다.

*거리 : 가로*세로 길이가 2.5cm인 ArUco마커를 기준으로 카메라와 마커간의 거리를 cm단위로 표시합니다.

웹캠을 통해 보이는 ArUco마커의 정보들을 확인할 수 있는 명령어 함수입니다.

다음과 같이 코딩 후 ArUco 마커의 정보를 직접 확인해 보세요.



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - Python 명령어 실습

```
1 from RoboCam.robocam import *
2 rCam = RoboCam()
3 rCam.WebcamStreamInit()
4 rCam.WebcamStream()
5 rCam.ArUcoDetectorInit()
6 rCam.ArUcoDetectorStart()
7 input()
```

웹캠을 향해

Original ArUco마커를 비추면 사진과 같이

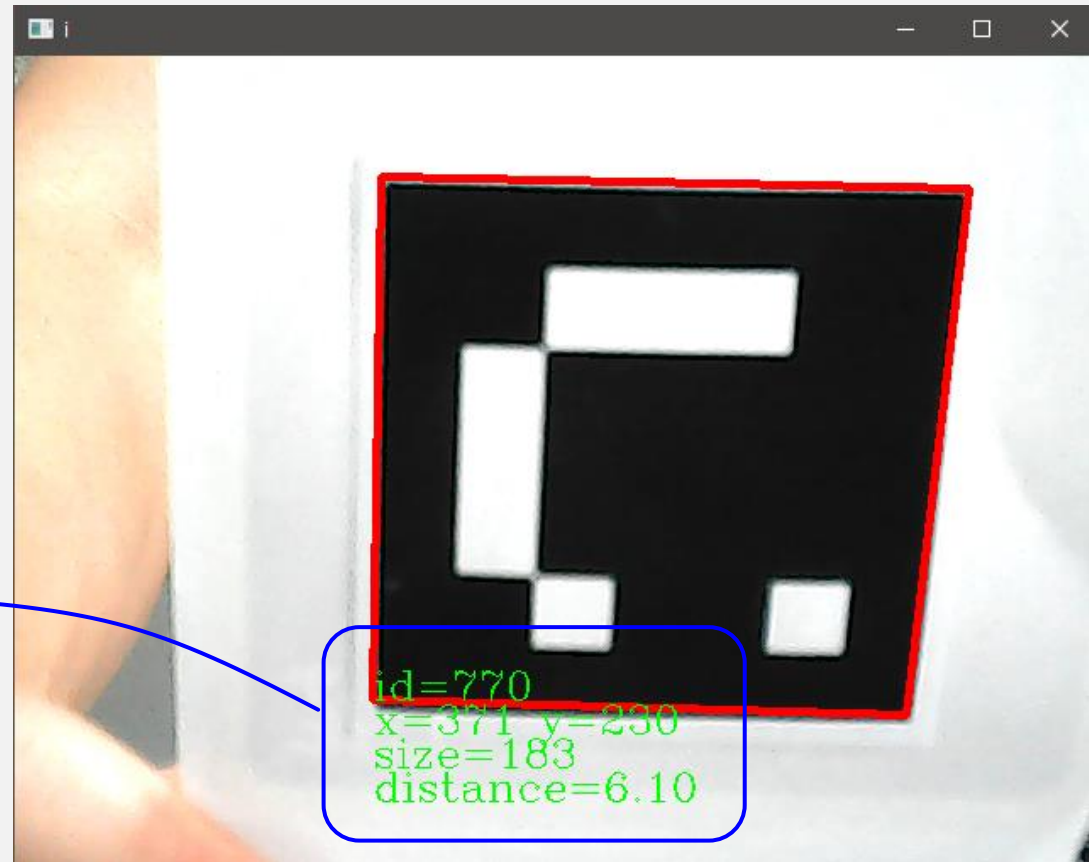
마커의 ID = 770

마커의 중심점 좌표 = 371,230

마커의 크기 = 183

마커와 화면과의 거리 = 6.10

정보가 순차적으로 노출됩니다.





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



코딩하기 - Python 명령어 실습

화면에 필요한 정보만을 노출시키기 위해서는 앞서 학습한 명령어들을 이용한 코딩을 진행해야 합니다.

```
Python test.py > ...  
1  from RoboCam.robocam import *  
2  rCam = RoboCam()  
3  rCam.WebcamStreamInit()  
4  rCam.WebcamStream()  
5  rCam.ArucoDetectorInit()  
6  rCam.ArucoDetectorStart()  
7  rCam.DrawArucoArea(False)  
8  rCam.DrawArucoId(False)  
9  rCam.DrawArucoPoint(False)  
10 rCam.DrawArucoDistance(False)  
11 rCam.DrawArucoSize(False)  
12 input()
```



기존 노출되었던
ArUco마커의 정보 및
마커의 외곽선을 숨길 수
있어요.





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - Python 명령어 실습

학습한 내용을 바탕으로, 다음 문제를 해결해 보세요.

1) 웹캠을 통해 보이는 ArUco 마커의 '윤곽선' '크기' 'ID번호' 만 표현되도록 코딩해 보세요!

```
Python test.py > ...
1  from RoboCam.robocam import *
2  rCam = RoboCam()
3  rCam.WebcamStreamInit()
4  rCam.WebcamStream()
5  rCam.ArucoDetectorInit()
6  rCam.ArucoDetectorStart()
7
8  Cam.DrawAruco ?
9
10 input()
```





Python으로 [워커] 제어하기

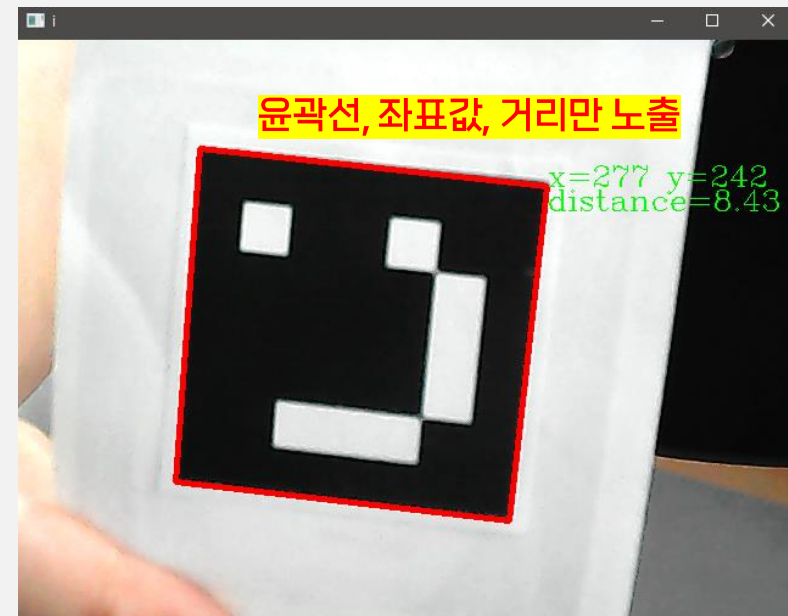
Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - Python 명령어 실습

학습한 내용을 바탕으로, 다음 문제를 해결해 보세요.

2) 웹캠을 통해 보이는 ArUco 마커의 '윤곽선' '좌표값' '거리' 데이터만 표현되도록 코딩해 보세요!

```
Python test.py > ...
1  from RoboCam.robocam import *
2  rCam = RoboCam()
3  rCam.WebcamStreamInit()
4  rCam.WebcamStream()
5  rCam.ArucoDetectorInit()
6  rCam.ArucoDetectorStart()
7
8  Can DrawAr ?
9  Can DrawArucoSize ?
10 input()
```





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



코딩하기 - Python 명령어 알기

스트리밍 된 화면에 보이는 ArUco마커의 정보를 데이터로 얻어, 프린트 하는 명령어를 알아볼까요?

GetArucold()->int

인식된 ArUco마커의 고유번호를 int형태로 반환하는 함수입니다.

GetArucoCount() -> int

인식된 ArUco마커의 개수를 반환하는 함수입니다.

GetArucoCenterPoint(id:int) -> list

인식된 ArUco마커의 중심좌표를 [x,y] 형태의 리스트로 반환하는 함수 입니다.

* id : 검출할 ArUco마커의 ID 입니다.

GetArucoX(id:int) -> int

인식된 ArUco마커의 개수를 반환하는 함수입니다. X좌표 데이터를 반환하는 함수입니다.

GetArucoY(id:int) -> int

인식된 ArUco마커의 개수를 반환하는 함수입니다. Y좌표 데이터를 반환하는 함수입니다.



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

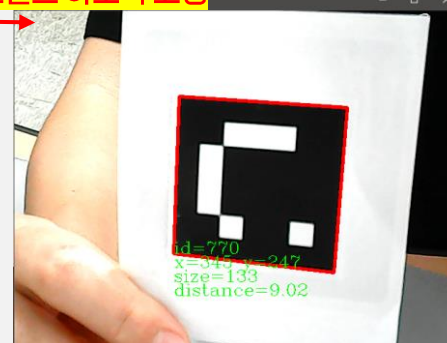
코딩하기 - Python 명령어 알기

학습한 코드를 사용해서, 터미널에 ArUco 마커의 정보를 프린트 해 보세요.

```
Python test.py > ...
1  from time import sleep
2  from RoboCam.robocam import *
3  rCam = RoboCam()
4  rCam.WebcamStreamInit()
5  rCam.WebcamStream()
6  rCam.ArUcoDetectorInit()
7  rCam.ArUcoDetectorStart()
8  while(True):
9      print("=====")
10     print(rCam.GetArUcoId())
11     print(rCam.GetArUcoCenterPoint(770))
12     print(rCam.GetArUcoX(770))
13     print("=====")
14     sleep(1)
```

기본함수 'Sleep'을 사용하기 위해 명령어 호출

반복문 내부에 1초에 한 번씩
ArUco 마커 정보를 프린트 하도록 코딩



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
770
[317, 299]
317
=====
770
[363, 260]
363
=====
770
[351, 252]
351
=====
```

마커번호
[x,y]좌표
크기



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - Python 명령어 알기

얼굴을 감지하고, 해당 데이터를 활용할 수 있는 명령어 함수에 대해 알아보까요?

FacedetectorInit()

- 웹캠의 얼굴인식 모듈을 준비하는 함수입니다.
(얼굴감지, 얼굴구별, 이목구비 인식을 초기화 합니다.)

FacedetectorStart()

- 얼굴감지 모듈을 실행하는 함수입니다.
- 스트리밍 화면 안에 들어온 얼굴을 인식하여 외곽선에 사각형을 표시합니다.
- 이목구비 좌표에 작은 원을 표시합니다.
- 학습된 얼굴 데이터가 있다면, 학습된 데이터와 비교하여 인식된 이름을 표시합니다.
- 얼굴감지 모듈을 중지하는 함수는 FacedetectorStop() 입니다.

다음페이지에 안내되는 내용을 따라, 웹캠으로 얼굴에 대한 데이터를 확인할 수 있어요.



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



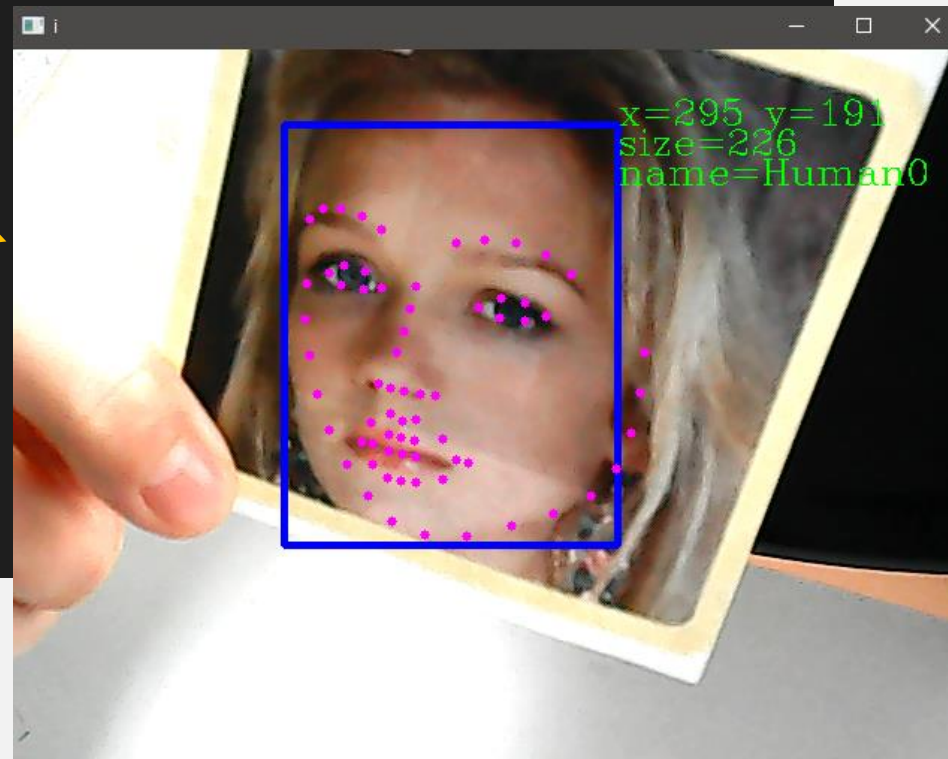
스트리밍화면 확인하기

Python test.py > ...

```
1 from time import sleep
2 from RoboCam.robocam import *
3 rCam = RoboCam()
4 rCam.WebcamStreamInit()
5 rCam.WebcamStream()
6 rCam.FacedetectorInit()
7 rCam.FacedetectorStart()
8 input()
```

실행시 스트리밍 화면 생성

터미널 창에 '엔터'키 입력시 스트리밍 종료





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



코딩하기 - Python 명령어 실습

웹캠을 향해

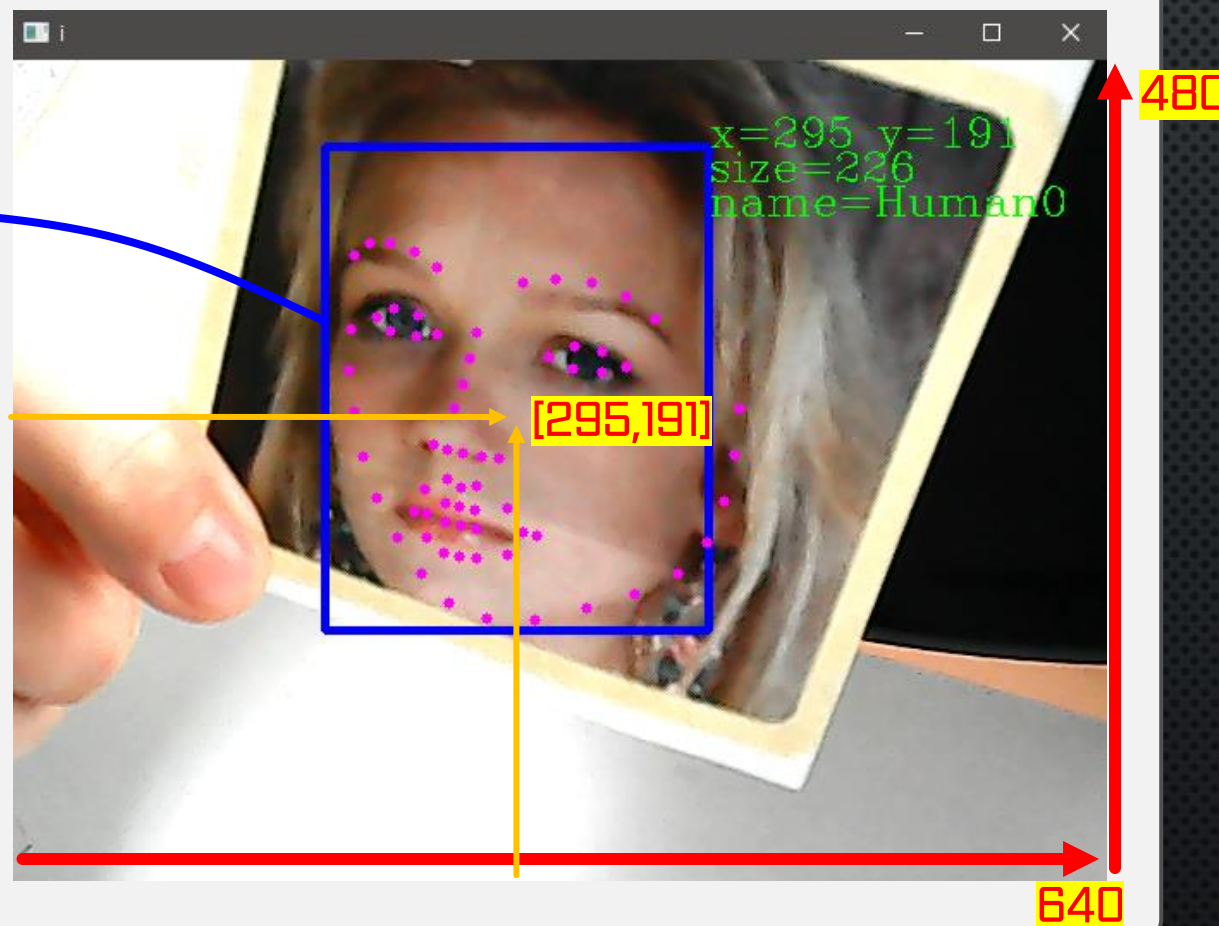
이목구비가 표현된 얼굴을 사진을 비추면,
사진과 같이

얼굴데이터의 중심좌표 : 295,191

얼굴 데이터의 크기 : 226

얼굴데이터의 이름 : Human0

정보가 순차적으로 노출됩니다.





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



코딩하기 - Python 명령어 알기

스트리밍 된 얼굴의 상세 데이터를 확인하는 명령어 함수에 대해 알아보까요?

DrawFaceArea(flag:bool)

- 화면에 인식된 얼굴의 영역 표시 여부를 정하는 함수입니다.
- * 아래부터 나열될 상세데이터 함수는 '영역표시'가 활성화 될 때 사용할 수 있어요.

DrawFacePoint(flag:bool)

- 화면에 인식된 얼굴의 좌표 표시 여부를 정하는 함수입니다.

DrawFaceSize(flag:bool)

- 화면에 인식된 얼굴의 좌표 표시 여부를 정하는 함수입니다.
- * 얼굴의 크기는 얼굴 윤곽선 사이의 거리(픽셀)를 말합니다.

GetFaceCount() -> int

- 화면에 인식된 얼굴의 개수를 반환하는 함수입니다.



Python으로 [워커] 제어하기

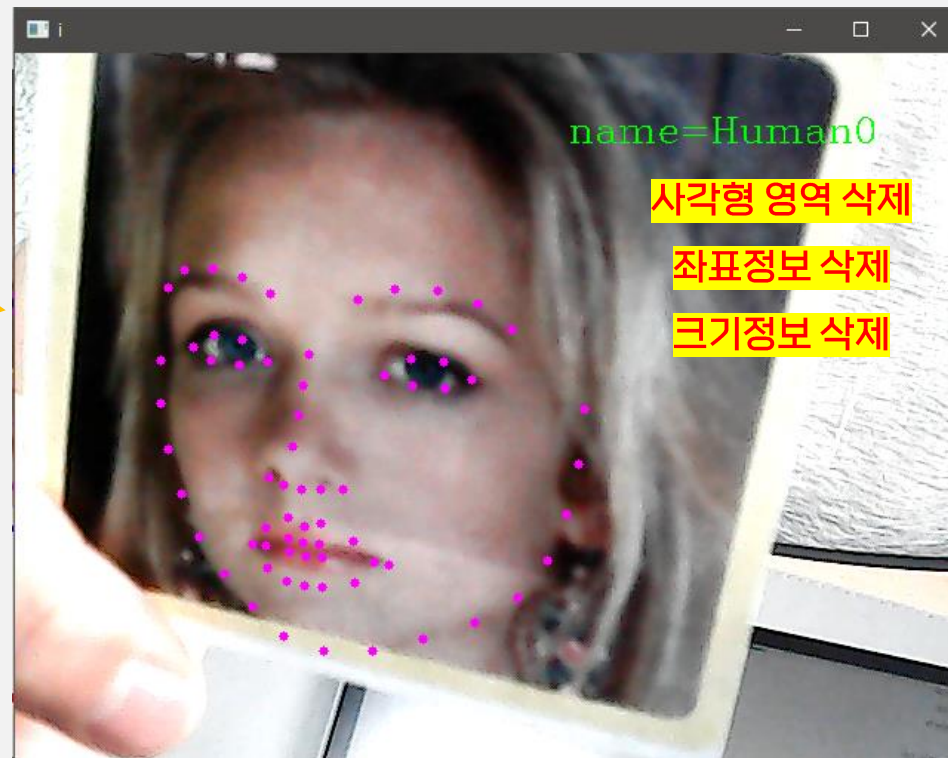
Visual Studio Code를 활용한 Python 인공지능 실습



스트리밍 화면 확인하기 (1)

```
Python test.py > ...  
1  ∨ from time import sleep  
2  from RoboCam.robocam import *  
3  rCam = RoboCam()  
4  rCam.WebcamStreamInit()  
5  rCam.WebcamStream()  
6  rCam.FacedetectorInit()  
7  rCam.FacedetectorStart()  
8  rCam.DrawFaceArea(False)  
9  rCam.DrawFacePoint(False)  
10 rCam.DrawFaceSize(False)  
11 input()
```

터미널 창에 '엔터'키 입력시 스트리밍 종료





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

스트리밍 화면 확인하기 (화면에 비친 얼굴 수 반환)

```
Python test.py > ...
1  from time import sleep
2  from RoboCam.robocam import *
3  rCam = RoboCam()
4  rCam.WebcamStreamInit()
5  rCam.WebcamStream()
6  rCam.FacedetectorInit()
7  rCam.FacedetectorStart()
8  while(True):
9      print(rCam.GetFaceCount())
10     sleep(1)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2
2
2
2
2
2

터미널 창에 인식된 얼굴의 숫자 프린트





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



코딩하기 - Python 명령어 알기

얼굴 데이터 학습 명령어에 대해 알아보까요?

FaceCapture(name:str, captureCount:int=5, path:str)

- 감지된 얼굴을 name 이름으로 captureCount횟수 만큼 캡처하여 path 위치에 저장합니다.
 - * name : 얼굴이 저장될 이름입니다. 같은 이름으로 여러 번 저장하면, 넘버링되어 저장됩니다.
 - * captureCount : 얼굴을 저장할 횟수 입니다. 0.1초 간격으로 지정된 횟수만큼 저장합니다.
 - * path : 얼굴 데이터가 저장될 경로입니다. 기본경로는 패키지 설치경로의 '/res/face/' 입니다.

TrainFaceData(path:str)

- facepath에 저장된 얼굴 데이터를 얼굴구별 모듈에 학습시키는 함수입니다.
학습된 데이터를 기반으로 웹캠에 확인된 얼굴의 이름이 표시됩니다.

DeleteFaceData(name:str, path:str)

- path에 저장된 name 이름을 가진 얼굴 데이터를 삭제하는 함수입니다. 해당 폴더에 접근하여 데이터를 삭제합니다.
데이터 삭제 이후 TrainFaceData 함수를 호출해야 얼굴구별 모듈에 적용됩니다.



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

스트리밍 화면 확인하기 (얼굴 데이터 학습하기 예제)

```
1 from RoboCam.robocam import *
2 rCam = RoboCam()
3 rCam.WebcamStreamInit()
4 rCam.WebcamStream()
5 rCam.FacedetectorInit()
6 rCam.FacedetectorStart()
```

영상인식 및 얼굴인식을 위한 기본설정

```
7 rCam.TrainFaceData()
8 while(True):
9     print("=====")
10    print("사용할 기능을 입력하세요")
11    print("1) capture")
12    print("2) train")
13    print("3) delete")
14    print("=====")
15    ins = input("명령어 입력: ")
```

지정 폴더 내 얼굴데이터 학습

```
16 if ins == 'train':
17     rCam.TrainFaceData()
18 elif ins == 'capture':
19     var = input("저장할 이름:")
20     rCam.FaceCapture(var)
21 elif ins == 'delete':
22     var = input("삭제할 이름:")
23     rCam.DeleteFaceData(var)
24 else:
25     print("잘못된 명령어입니다. 다시입력하세요")
```

Input() 명령어를 통해 외부입력을 받기 전까지

좌측과 같이 6줄의 내용을 프린트

입력받은 값을 ins 값으로 지정

입력받은 값 'ins'가

train 일 경우 TrainFaceData()함수 실행

capture일 경우 외부입력을 받아 FaceCapture(var)함수 실행

delete일 경우 외부입력을 받아 DeleteFaceData(var)함수 실행

입력 받은 값이 위 3항이 아닌 경우, "잘못된 명령어입니다. 다시

입력하세요" 출력



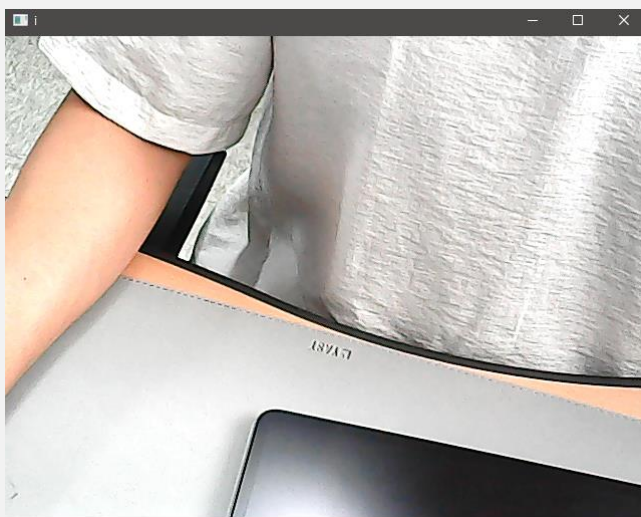
Python으로 [워크] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



스트리밍 화면 확인하기 (얼굴 데이터 학습하기 실습 1)

1) 프로그램 실행 시 웹캠 오픈 & 터미널을 통해 미리 코딩 된 문구 프린팅



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
=====
사용할 기능을 입력하세요
1) capture
2) train
3) delete
=====
명령어 입력: 
```

```
while(True):
    print("=====")
    print("사용할 기능을 입력하세요")
    print("1) capture")
    print("2) train")
    print("3) delete")
    print("=====")
    ins = input("명령어 입력: ")
```



Python으로 [워크] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

스트리밍 화면 확인하기 (얼굴 데이터 학습하기 실습 1)

2) 아래 설명을 따라 얼굴 데이터 저장하기

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

=====
사용할 기능을 입력하세요
1) capture
2) train
3) delete
=====
명령어 입력: capture
```



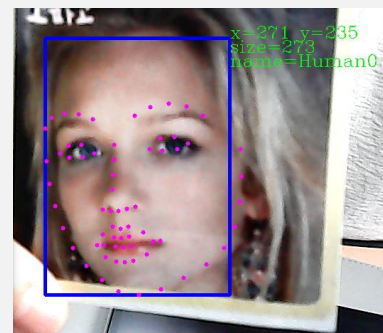
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

사용할 기능을 입력하세요
1) capture
2) train
3) delete
=====
명령어 입력: capture
저장할 이름: kim
```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

사용할 기능을 입력하세요
1) capture
2) train
3) delete
=====
명령어 입력: capture
저장할 이름: kim
```



웹캠에 학습시킬
얼굴을 위치시킴



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

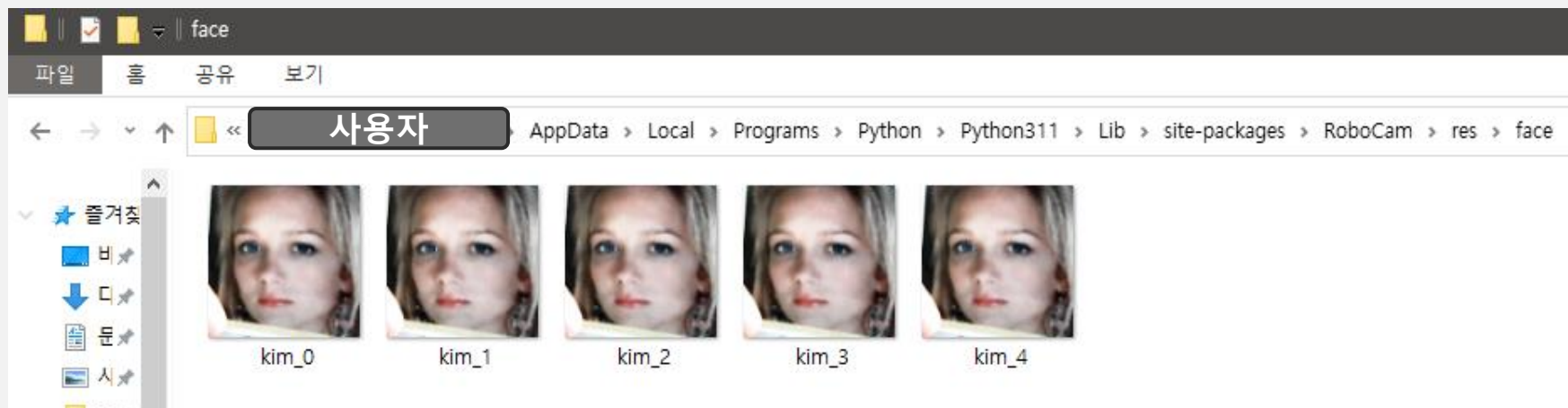


스트리밍 화면 확인하기 (얼굴 데이터 학습하기 실습 1)

3) 촬영된 데이터 확인하기

학습된 데이터를 확인하고 싶다면, 아래 경로에서 확인이 가능합니다.

C:\User > AppData > Local > Programs > Python > Python310 > Lib > site-packages > RoboCam > res > face
or
Python311



위와 같이 5장의 사진이 찍혀있는 걸 확인할 수 있어요.



Python으로 [워크] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



스트리밍 화면 확인하기 (얼굴 데이터 학습하기 실습 1)

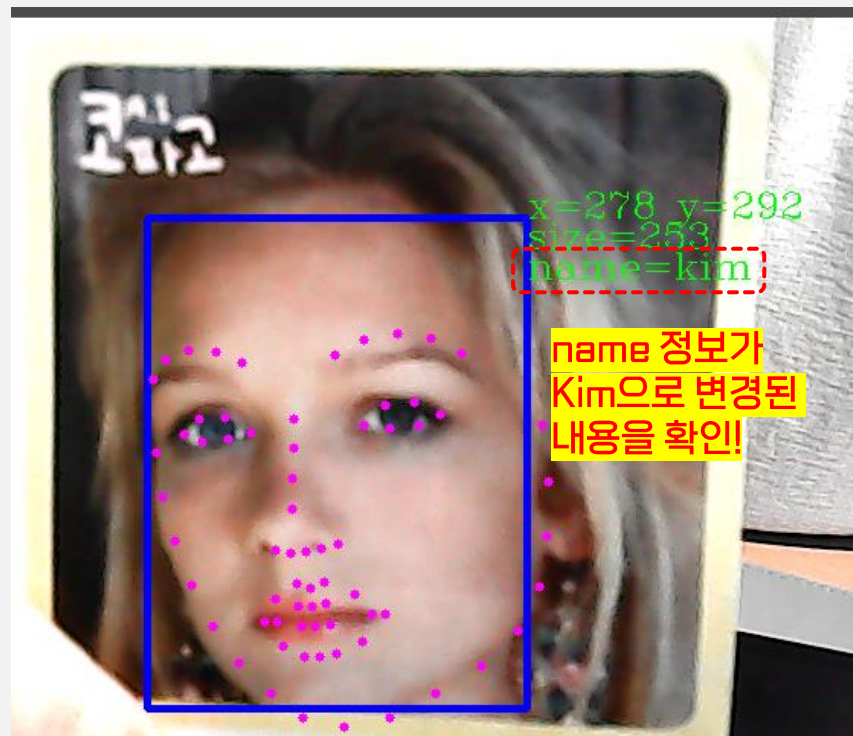
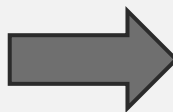
4) 촬영된 데이터 학습하기

```
PROBLEMS    OUTPUT    DEBUG CONSOLE

2) train
3) delete
=====
명령어 입력: capture
저장할 이름: kim
kim is saved
=====
사용할 기능을 입력하세요
1) capture
2) train
3) delete
=====
명령어 입력: train
```

train 명령어 입력 후
'엔터' 입력

train 명령어 입력 후
웹캠에 다시
사진을 비춰보면..





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



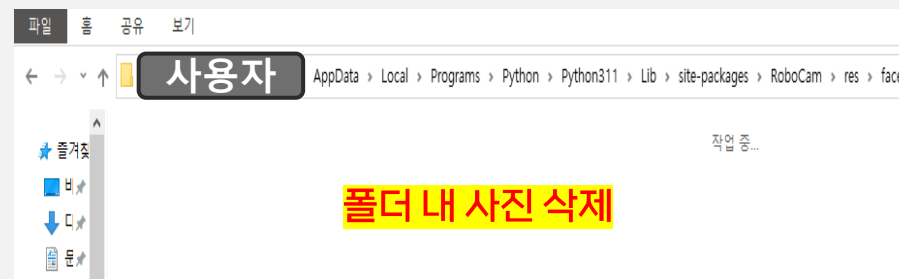
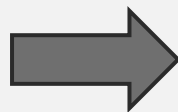
스트리밍 화면 확인하기 (얼굴 데이터 학습하기 실습 1)

5) 촬영된 데이터 삭제하기

```
PROBLEMS  OUTPUT  DEBUG CONSOLE

=====
사용할 기능을 입력하세요
1) capture
2) train
3) delete
=====
명령어 입력: train
=====
사용할 기능을 입력하세요
1) capture
2) train
3) delete
=====
명령어 입력: delete
삭제할 이름: kim
```

delete 명령어 입력
학습한 이름 입력하면,



delete 명령어 입력
-> 삭제할 이름에 kim 입력
-> '엔터' 입력



Python으로 [워크] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

스트리밍 화면 확인하기 (얼굴 데이터 학습하기 실습 1)

6) 학습하기를 통해 데이터 새로고침

PROBLEMS OUTPUT DEBUG CONSOLE

삭제할 이름:kim

kim is deleted

=====

사용할 기능을 입력하세요

1) capture

2) train

3) delete

=====

명령어 입력: train

train명령어 입력 후
'엔터' 입력

=====

사용할 기능을 입력하세요

1) capture

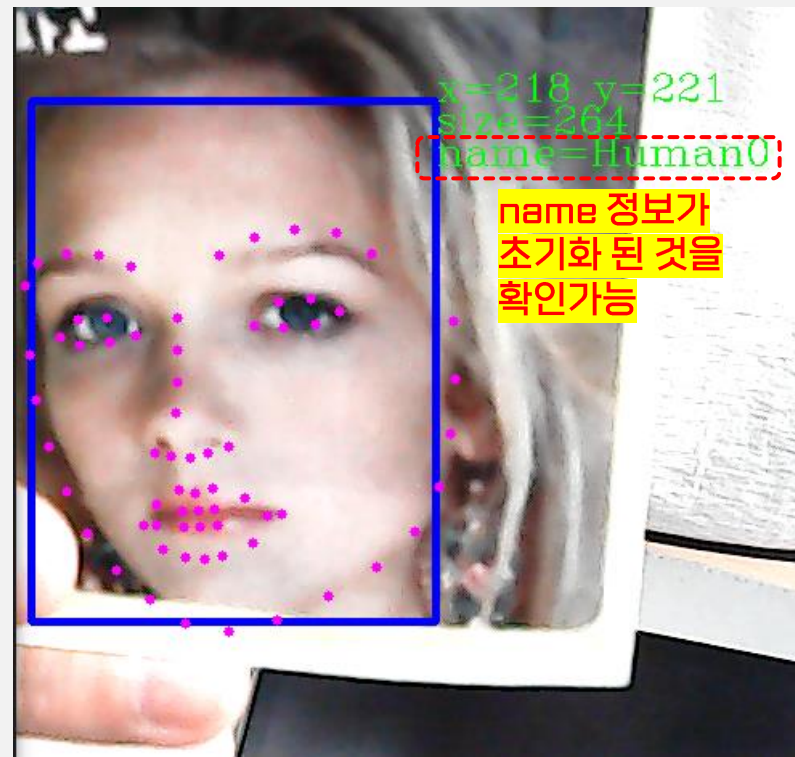
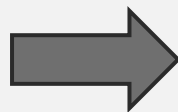
2) train

3) delete

train 명령어 입력 후

웹캠에 다시

사진을 비춰보면..





Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

코딩하기 - Python 명령어 알기

스트리밍 된 화면에 보이는 얼굴 정보를 데이터로 얻어, 프린트 하는 명령어를 알아볼까요?

GetFaceNames() -> list

인식된 얼굴 중 name의 이름을 가진 얼굴이 있는지 반환하는 함수입니다.

(인식된 얼굴이 없다면 빈 리스트를 반환)

GetFaceSize(name:str) -> int

인식된 얼굴 중 name의 이름을 가진 얼굴의 크기를 반환하는 함수 입니다.

GetFaceCenterPoint(name:str) -> list

인식된 얼굴 중 name의 이름을 가진 얼굴의 중심좌표를 반환하는 함수입니다.

GetFaceExist(name:str) -> bool

인식된 얼굴 중 name의 이름을 가진 얼굴이 있는지 여부를 반환하는 함수입니다.

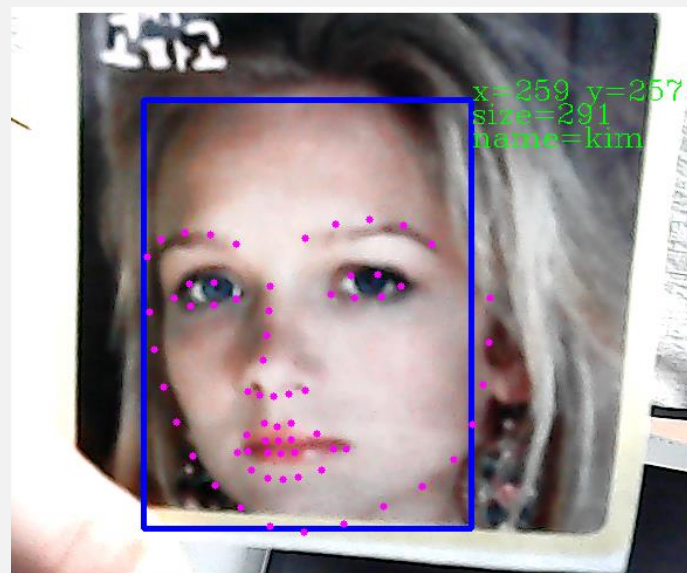


Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

스트리밍 화면 확인하기 (kim에 대한 데이터 사전학습 진행)

```
Python test.py > ...
1  from time import sleep
2  from RoboCam.robocam import *
3  rCam = RoboCam()
4  rCam.WebcamStreamInit()
5  rCam.WebcamStream()
6  rCam.FacedetectorInit()
7  rCam.FacedetectorStart()
8  rCam.TrainFaceData()
9  while(True):
10     print("=====")
11     print(rCam.GetFaceNames())
12     print(rCam.GetFaceExist('kim'))
13     print(rCam.GetFaceSize('kim'))
14     print(rCam.GetFaceCenterPoint('kim'))
15     print("=====")
16     sleep(1)
```



```
=====
['kim']
True
291
[261, 240]
=====
```

등록된 이름
'kim'의 존재여부
'kim'의 크기
'kim'의 중심점 좌표



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



코딩하기 - Python 명령어 알기

스트리밍 된 화면에 보이는 얼굴 데이터의 이목구비 정보 반환 함수 알아보기 (I)

```
GetLeftIrisPoint(name:str)->list
```

인식된 얼굴의 왼쪽 눈동자 좌표를 반환하는 함수 입니다.

```
GetLeftEyebrowPoint(name:str)->list
```

인식된 얼굴의 왼쪽 눈썹 좌표를 반환하는 함수 입니다.

```
GetRightIrisPoint(name:str)->list
```

인식된 얼굴의 오른쪽 눈동자 좌표를 반환하는 함수 입니다.

```
GetRightEyebrowPoint(name:str)->list
```

인식된 얼굴의 오른쪽 눈썹 좌표를 반환하는 함수 입니다.



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



코딩하기 - Python 명령어 알기

스트리밍 된 화면에 보이는 얼굴 데이터의 이목구비 정보 반환 함수 알아보기 (2)

```
GetNosePoint(name:str)->list
```

인식된 얼굴의 코 좌표를 반환하는 함수 입니다.

```
GetMousePoint(name:str)->list
```

인식된 얼굴의 입 좌표를 반환하는 함수 입니다.

```
GetJawPoint(name:str)->list
```

인식된 얼굴의 턱 좌표를 반환하는 함수 입니다.

```
DrawFaceLandmark(flag:bool)
```

인식된 얼굴의 이목구비 포인트 표시 여부를 지정하는 함수입니다.

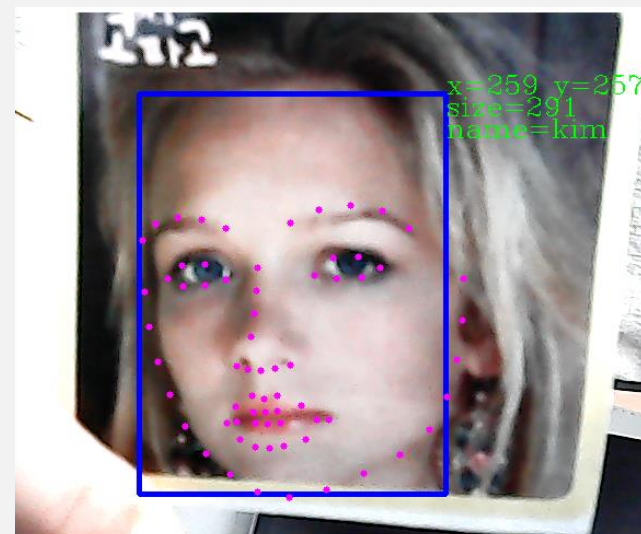


Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

스트리밍 화면 확인하기

```
Python test.py > ...
1  from time import sleep
2  from RoboCam.robocam import *
3  rCam = RoboCam()
4  rCam.WebcamStreamInit()
5  rCam.WebcamStream()
6  rCam.FacedetectorInit()
7  rCam.FacedetectorStart()
8  rCam.TrainFaceData()
9  while(True):
10     print("=====")
11     print(rCam.GetLeftIrisPoint('kim'))
12     print(rCam.GetLeftEyebrowPoint('kim'))
13     print(rCam.GetNosePoint('kim'))
14     print(rCam.GetMousePoint('kim'))
15     print(rCam.GetJawPoint('kim'))
16     print("=====")
17     sleep(1)
```



```
=====
=====
[186.0, 225.0]
[169, 170]
[240, 305]
[257.5, 348.5]
[264, 417]
=====
```

1초마다
왼쪽 눈동자
왼쪽 눈썹
코
턱
좌표를 프린트



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습

카메라 인식 데이터를 통해 [워커] 제어하기

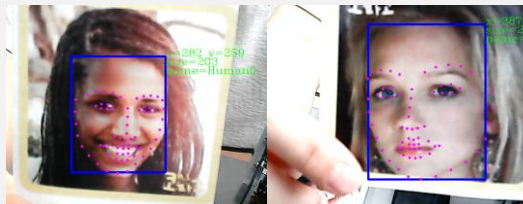
```
1 from RoboCam.robocam import *
2 from RobokitRS import *
3 from time import sleep
4 rCam = RoboCam()
5 rCam.WebcamStreamInit()
6 rCam.WebcamStream()
7 rCam.FacedetectorInit()
8 rCam.FacedetectorStart()
9 name1 = input("Name1 : ")
10 rCam.FaceCapture(name1)
11 sleep(1)
12 name2 = input("Name2 : ")
13 rCam.FaceCapture(name2)
14 sleep(1)
15 rCam.TrainFaceData()
16 rs = RobokitRS.RobokitRS()
17 rs.port_open("COM5")
18 while True:
19     if rCam.GetFaceExist(name1):
20         rs.servo_write(2,90)
21         sleep(0.5)
22         rs.servo_write(2,70)
23         sleep(0.5)
24         rs.servo_write(2,90)
25         sleep(0.5)
26     elif rCam.GetFaceExist(name2):
27         rs.servo_write(4,-90)
28         sleep(0.5)
29         rs.servo_write(4,-70)
30         sleep(0.5)
31         rs.servo_write(4,-90)
32         sleep(0.5)
33     sleep(0.2)
```

웹캠 및 로봇제어 라이브러리 호출

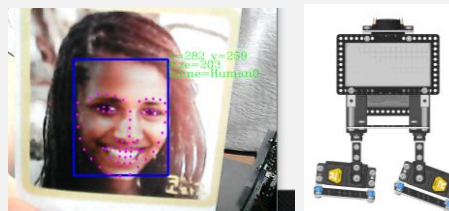
얼굴인식 라이브러리 세팅

2개의 얼굴을 순차적으로 학습

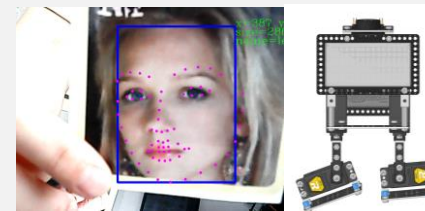
학습된 2개의 얼굴을 볼 때마다 코딩 된 동작 실행



```
Name1 : kim
kim is saved
Name2 : lee
lee is saved
```



Kim 인식할 때
왼쪽다리 움직임



lee 인식할 때
오른쪽다리 움직임



Python으로 [워커] 제어하기

Visual Studio Code를 활용한 Python 인공지능 실습



영상인식을 활용해 Python으로 서보모터 제어하기

얼굴 정보를 등록하고, 각 얼굴을 인식했을 때 별도의 움직임을 가지도록 코딩해 보세요.

