**Monitor.py**

**vk\_human\_detect(Interval):**

Use visionkit to detect human,when people are recognized,get screenshot(image/jpg) from visionkit,then send sensor data and image to Slack

**\*you can use “for” to start a circle detect**

while(True):

vk\_human\_detect(10)

**\*you can change**

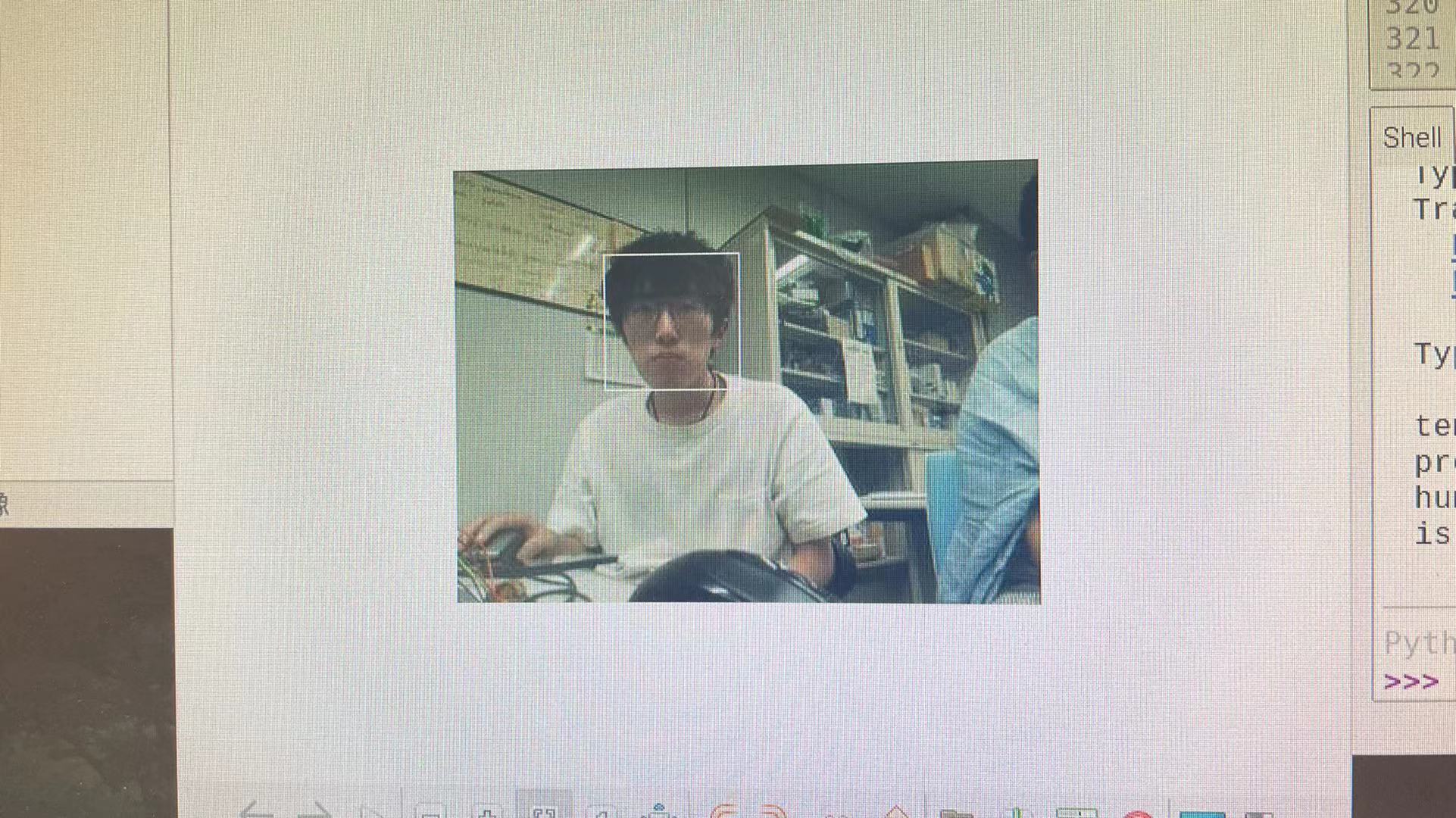
out=vk.vk\_recog\_face(picname)#no rectangle

**into**

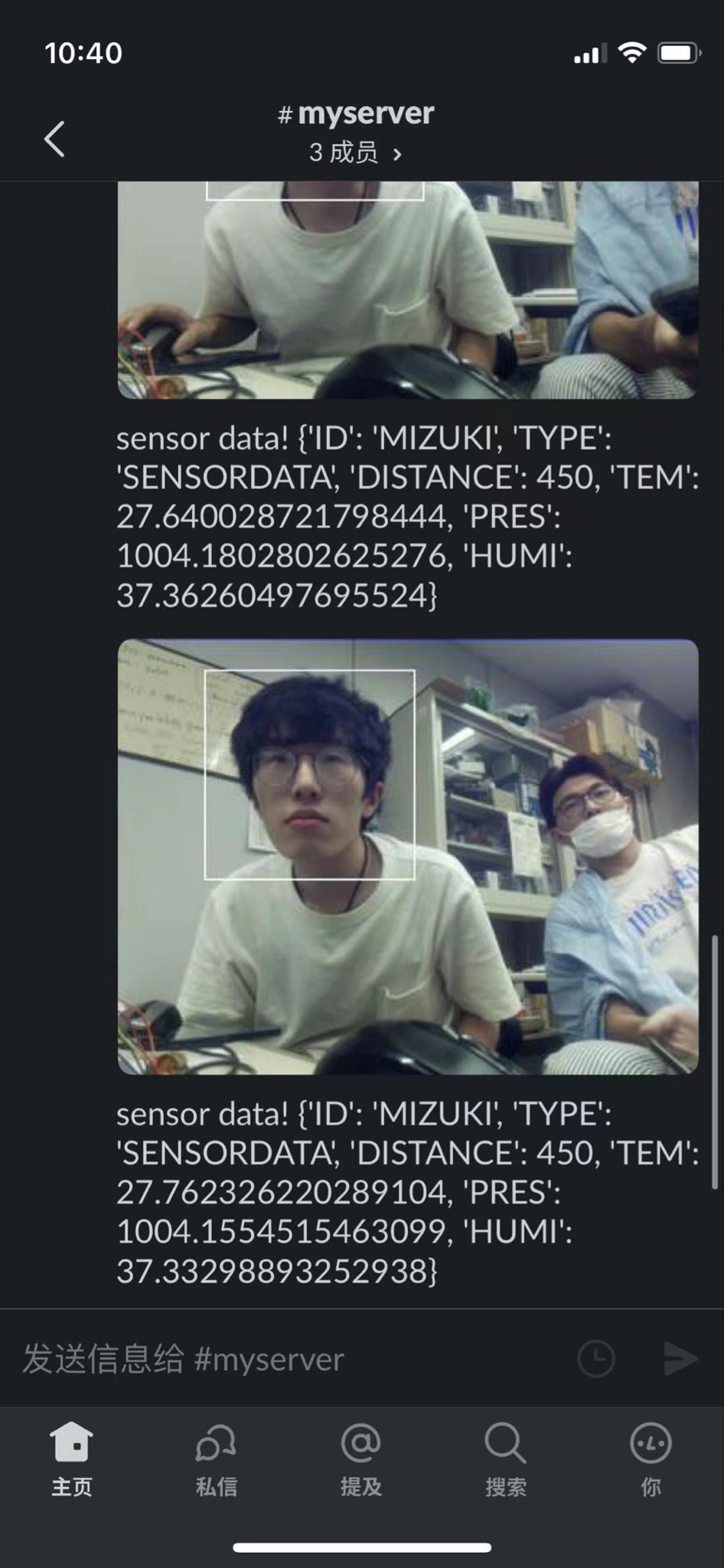
out=vk.vk\_box\_face(picname)#have rectangle

**to start a rectangle-have detection**

the image got by vk.vk\_box\_face(picname) will be like this:



in slack,the data will be like this



with a rectangle!(vk\_box\_face)

**Visionkit.py**

**def vk\_box\_stream(self,picname):**

start a human detect stream in visionkit(lasting but no scrennshot)

**def vk\_box\_face(self,picname):**

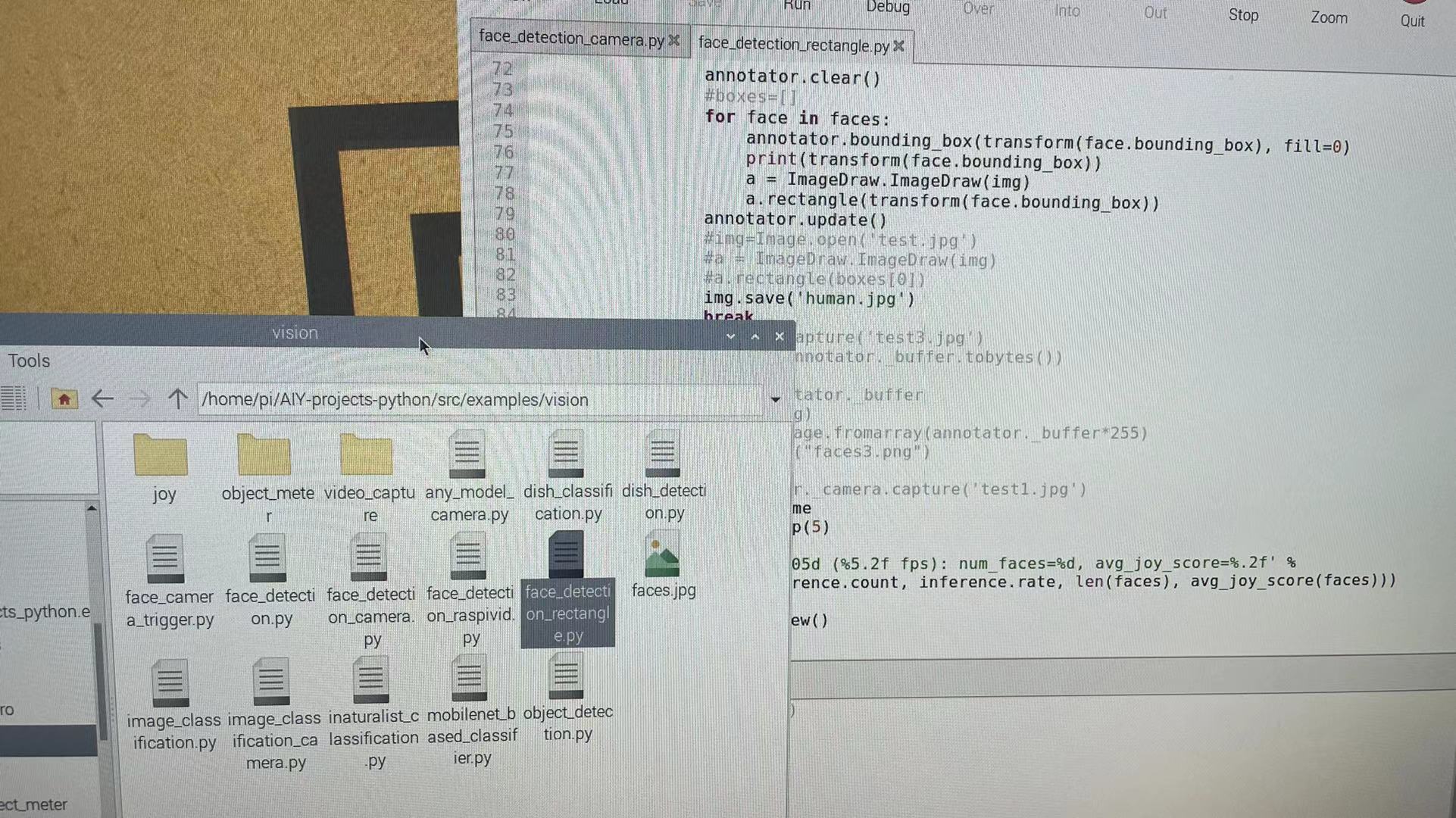
start a human detect in visionkit(take a scrennshot with rectangle and save to PI)

**\*if you want to use vk\_box\_face.py**

**in visionkit**(~AIY-projects-python/src/examples/vision/)

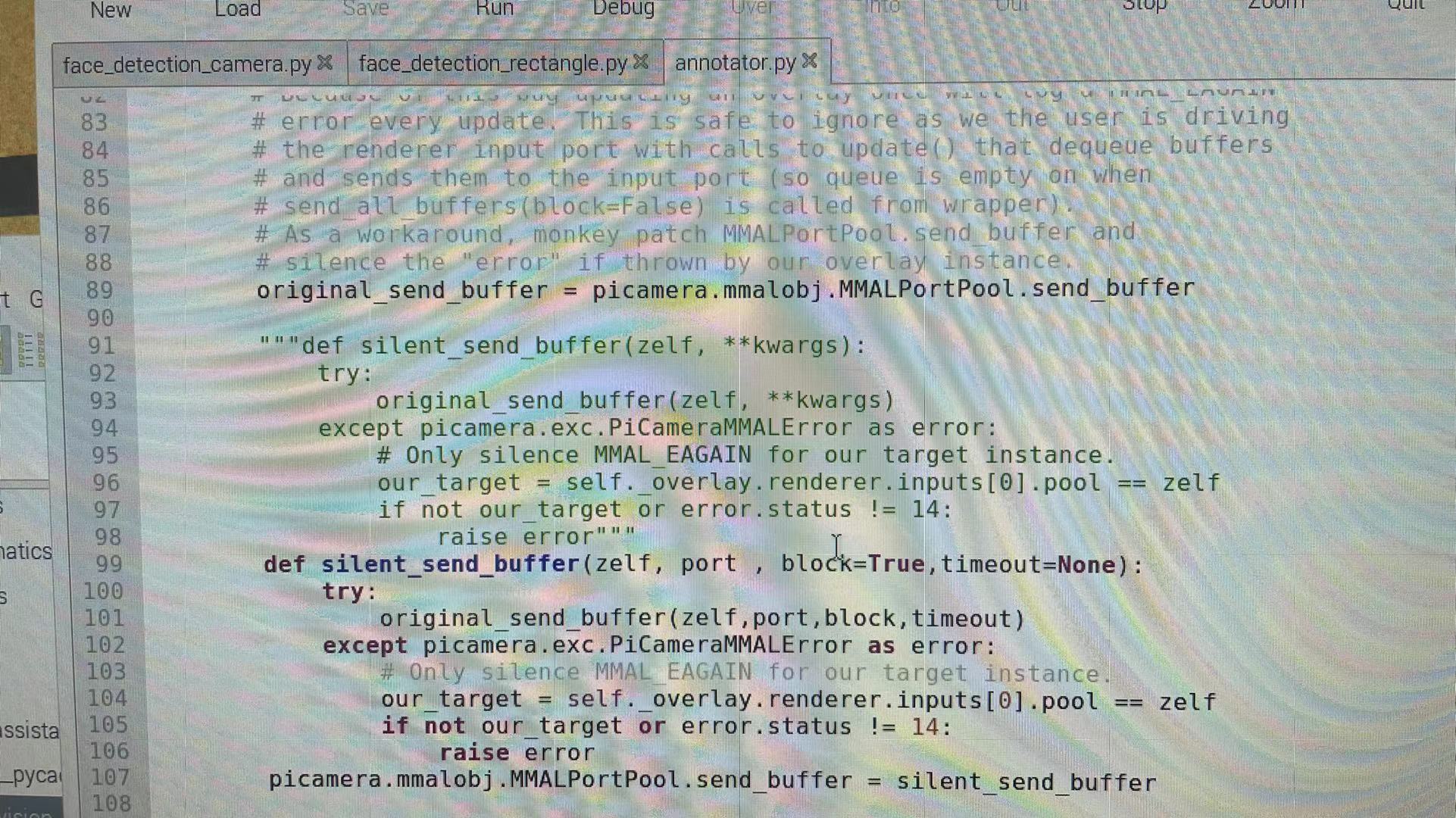
**you should put face\_detection\_rectangle.py here(like photo below)**

(you can find face\_detection\_rectangle.py in our ラズベリーパイ/human\_detect/AIY/)



**and in visionkit**(~AIY-projects-python/src/aiy/vision/)

**you must replace annotator.py**(~AIY-projects-python/src/aiy/vision/) **like this:**

****

visionkit's source code has a bug that can not screenshot(capture) while buffer\_streaming

if you change code like this,you can make it and run face\_detection\_rectangle.py correctly

**annotator.py changed code:**

original\_send\_buffer = picamera.mmalobj.MMALPortPool.send\_buffer

"""def silent\_send\_buffer(zelf, \*\*kwargs):

try:

original\_send\_buffer(zelf, \*\*kwargs)

except picamera.exc.PiCameraMMALError as error:

# Only silence MMAL\_EAGAIN for our target instance.

our\_target = self.\_overlay.renderer.inputs[0].pool == zelf

if not our\_target or error.status != 14:

raise error"""

def silent\_send\_buffer(zelf, port , block=True,timeout=None):

try:

original\_send\_buffer(zelf,port,block,timeout)

except picamera.exc.PiCameraMMALError as error:

# Only silence MMAL\_EAGAIN for our target instance.

our\_target = self.\_overlay.renderer.inputs[0].pool == zelf

if not our\_target or error.status != 14:

raise error

picamera.mmalobj.MMALPortPool.send\_buffer = silent\_send\_buffer

**or you can find annotator.py at our file** (ラズベリーパイ/human\_detect/AIY/)

**def vk\_recog\_face(self,picname):**

start a human detect in visionkit(take a scrennshot without rectangle and save to PI)

**def vk\_human\_detect(self,interval):**

start a “for circle” human-detection(when detect a human ,scrennshot and save image to PI)

**Slacktest.py/slack\_photo.py**

test slack function

**Document /BME280**

the dependence lib that can run Bme280 sensor to get environment data

**Document /AIY**

in this document,there are python codes that you should put them into Visionkit

So you can realize something new