Article
Yuri Marx · Jan 31, 2022 8m read

Open Exchange

Object Detection using Embedded Python and IRIS

From IRIS 2021.2 is possible write Class Methods using the Python Language. I used this new feature to detect persons and objects into images, using ImageAI (https://github.com/OlafenwaMoses/ImageAI). The ImageAI creator defines it as: "An open-source python library built to empower developers to build applications and systems with self-contained Deep Learning and Computer Vision capabilities using simple and few lines of code." In this article you will learn how to apply AI Computer Vision to detect object and persons inside images.

Steps to analyze images using ImageAI

1. Go to https://openexchange.intersystems.com/package/AI-Image-Object-Detector and click Download button to go to the Git project.
2. Clone/git pull the repo into any local directory

```
$ git clone https://github.com/yurimarx/image-analyzer.git
```

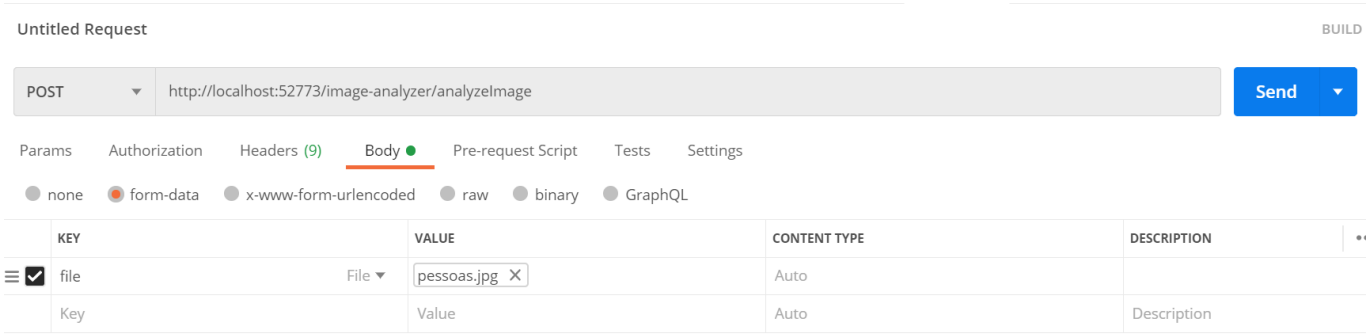
2. Open a Docker terminal in this directory and run:

```
$ docker-compose build
```

3. Run the IRIS container:

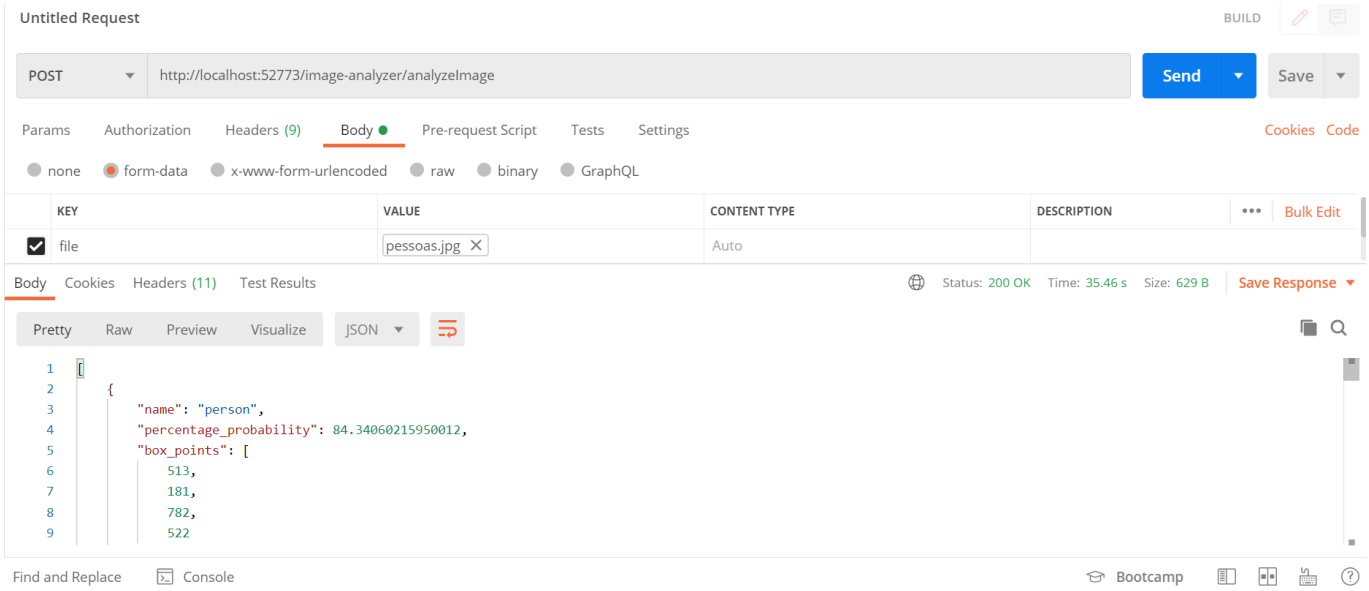
```
$ docker-compose up -d
```

4. Go to your Postman (or other similar REST client) and config the request like this image:



- Method: POST
- URL: http://localhost:52773/image-analyzer/analyzeImage
- Body: form-data
- Key: file (the name of file field must be file) and type File
- Value: select an image with persons and/or objects from your computer

5. Click send and get a response with detected objects like this:



Behind the scenes - the source code

The Dockerfile

It is an import step. To use python libraries you need to install it before, but you must pay attention to install inside the correct Python IRIS folder (/usr/irissys/mgr/python):

Dockerfile

- pip3 is the python tool used to install python libraries, like imageai, opencv, tensorflow and other.
- target parameter was used to install the python libraries where IRIS requires, the folder /usr/irissys/mgr/python.
- ImageAI uses the following python libraries:
    - TensorFlow: "is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++" (source: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine...)
    - Keras: "Keras is a deep learning API written in Python, running on top of the machine learning

platform TensorFlow. It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result as fast as possible is key to doing good research."* (source: https://keras.io/about/)
    - OpenCV: "(Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products" (source: https://opencv.org/about/).
- ImageAI is a framework that combines TensorFlow, Keras and OpenCV to allows you train and/or use market deep learning models easily to:
    - Image prediction;
    - Object Detection;
    - Video object detection and traking;
    - Video analysis;
    - Custom model training (imagine train a model to detect risk of patient falling out of bed in hospital ICU, for example).
- ImageAI works with the 3 main Computer Vision models used in the market for object detection:
    - RetinaNet *(Size = 145 mb, high performance and accuracy, with longer detection time)*
    - YOLOv3 *(Size = 237 mb, moderate performance and accuracy, with a moderate detection time)*
    - TinyYOLOv3 *(Size = 34 mb, optimized for speed and moderate performance, with fast detection time)*
- The input folder stores the images to be analyzed;
- The output folder stores the results (images with visual result tags);
- The models folder stores the trained models to be used with imageai.
- This sample uses RetinaNet model (restnet50)

Embedded Python implementation

The source code implementation is simple, and a few lines we do object detection (because ImageAI). If tensorflow was used directly with OpenCV and Keras, many more lines of code would have been written. See the commented code:

ImageAI implementation

The IRIS API to expose the Object Detection as an IRIS Object Detection Microservice

It is very easy to call the python classmethod, is it similiar to call any objectscript classmethod, see:

IRIS microservice to detect objects inside images

You can do many things with Embedded Python and ImageAI. Know more about ImageAI into: https://github.com/OlafenwaMoses/ImageAI.

This sample app using imageai will be participate in the Python contest, if you liked, vote it. Thanks!

#Embedded Python #Python #InterSystems IRIS
Check the related application on InterSystems Open Exchange