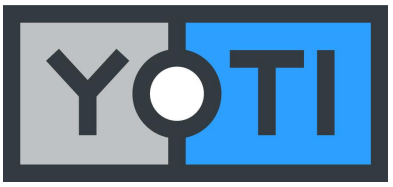
AI Services Technical Task



Intro

At Yoti we use a microservice architecture. This test asks for the development of an image classification service to fit this architecture. In AI Services we use TensorFlow Serving to serve our machine learning models.

For example, if we wanted to perform face detection on an image we would have two services. A face detection service which receives the image from the client and the TensorFlow Serving service which is running the face detection model, the face detection service would forward the image to the Tensorflow Serving image which would in turn return the data which contains the bounding boxes and landmarks.

Task

Please implement an image classification service using Docker. This service should have an endpoint which accepts an image and returns the classification for that image. To perform the classification please use the following public ResNet [model]
Note: This service should be implemented in Python 3.

There should be two services:

- Image classification service which receives the request from the client.
- Tensorflow Serving service which is running the ResNet model for image classification.

Image Classification service

It should retrieve the classification for the image using the ResNet model service and return the classification back to the client.
Inputs:

- Image

Outputs:

- Image classification

Tensorflow Serving

Import the model into a Tensorflow Serving Docker image. The model provides the following input and outputs:
Inputs:

- Image

Outputs:

- The output from this model will contain the probabilities and classes data. We only need the classes returned to the client. Example JSON response from the model:

  {'predictions': [{'probabilities': [...,...]}],'classes': 286}

Extra credits

Optional for extra credits! Only do these if you wish to spend more time on this. We're happy to consider answers with and without these.

1. Include unit testing.
2. Docker-compose file to run and test the services.

Notes

All work should be committed to git repo where the commit history can be reviewed. github is fine, but we're also happy for a .tar or similar of the git repository. Your solution should also be easy to run/verify. To that end please feel free to provide any further instructions, documentation, etc on how to go about the verification process.

Links

- [TensorFlow Serving with docker](https://www.tensorflow.org/tfx/serving/docker)
- [TensorFlow Serving REST API](https://github.com/tensorflow/serving/blob/master/tensorflow_serving/g3doc/api_rest.md)