Python Software Engineer Test Task

Many photographers have been taking images of birds and wondering what kind of bird it actually is.

A bunch of data scientists have been working on a model to help them out.

While the model* is performing well a lot of corners were cut to get this model to production** and the service could certainly use some love from a software engineer.

Your task is to:

- Improve service architecture
- Improve service performance
- Improve service maintainability, extendability and testability

You can change all parts of the code as you see fit, however:

- You are not expected to work on ML model performance
- Model and data have to be fetched online (instead of downloading it to your local machine)

By the end of this task we would like to see, what is a good-looking code in your opinion and how much can you optimize latency.

Feel free to play around with the code as much as you like, but in the end, we want to see:

- Your vision of nice code
- Code running time including images and model downloading and model inference
- Top 3 results from the model's output per image
- Proper logging for essential and debug info if necessary
- Finished work has to be pushed to GitHub and shared with @rivol, @khadrawy, and @suur

Bonus

- Unit tests with Mocked images and model data (possible to run without internet)
- Analyze the bottlenecks in your implementation, and report options for improving upon them.
- Implement your solution using Docker and Kubernetes for the infrastructure layer. The configuration should scale out: adding machines should reduce latency

Local setup

1. Install Python 3
2. Install requirements `pip install -r requirements.txt`
3. Run the code `python classifier.py`

gl;hf

* The model:

The sample model is taken from Tensorflow Hub:

TensorFlow Hub

The labels for model outputs can be found here:

https://www.gstatic.com/aihub/tfhub/labelmaps/aiy_birds_V1_labelmap.csv

The model has been verified to run with TensorFlow 2.

** Production: The code was deployed as a python service using Docker with Kubernetes for the infrastructure layer.

In case of questions feel free to contact Agu Suur at agu.suur@veriff.net