UDACITY

‹ Return to "Intel® Edge AI for IoT Developers Nanodegree" in the classroom

DISCUSS ON STUDENT HUB

Deploy a People Counter App at the Edge

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Overall,

Excellent work. Wish you all the best for your next project

Inference

A link to the original model is included, along with the command used in the terminal to convert it to an Intermediate Representation with the Model Optimizer. This can be noted in the project README or in the Submission Details section for the reviewer (on the submission page).

The model *cannot* be one of the existing Intermediate Representations provided by Intel®.

This should be included irrespective of the additional option to use one of the IR models if a suitably accurate model is not found (see Write-Up section).

Dear student,

Excellent start for the project

In `inference.py`, a person detection model is loaded using the Inference Engine.

Model loaded perfectly!

`inference.py` includes code that will notify the user that certain layers are not supported (if applicable), and allows for the use of a CPU extension in this case.

Layers are checked and extension added

Requests for inference are handled asynchronously through the Inference Engine.

Perfect Job!

The output of the network is appropriately returned through `get_output()`.

Well done!

Processing Video

In `main.py`, an input image or video file is handled appropriately, along with any potential webcam or similar streaming video.

The user is notified if a given input is unsupported.

Different forms of input are handled properly. Nice work

Appropriate preprocessing has been performed on an input image frame for input into the inference network.

Preprocessing completed as per the need.

The output of the network is appropriately processed, with bounding boxes/semantic masks extracted from the output, as applicable.

If using bounding boxes, it utilizes a probability threshold to toss out low confidence detections.

Excellent job!

Statistics regarding people on screen, duration they spend on screen, and total people counted are calculated based off of the detected bounding boxes/semantic masks.

The logic above results in close to actual statistics (no more than +/- 1 on people or +/- 2 seconds for time spent on screen).

Sending Data to Servers

The calculated statistics are sent through JSON to a MQTT server.

In the classroom workspace, this should be on port 3001 and publish to topics `person` and `person/duration`. `person` should have the keys `total` for the total count and `count` for the current count, while `person/duration` should use a `duration` key for publishing information.

Data flushed to server perfectly

The image frame, including any drawn outputs, are flushed to a FFmpeg server.

The video and statistics are viewable through a UI. This can either use the provided UI or one created by the student.

Write-Up

A write-up is provided that includes an explanation of the process behind converting any custom layers, as well as explaining the potential reasons for handling custom layers in a trained model.

Explained nicely.

Create and explain your method behind comparing the performance of a model with and without the use of the OpenVINO™ Toolkit (accuracy, size, speed, CPU overhead).

Also, compare the differences in network needs and costs of using cloud services as opposed to deploying at the edge.

There is around 7% increase in speed along with good reduction on model size. This is inline with the expected result.

Explain potential use cases of a people counter app.

This should include more than just listing the use cases - explain how they apply to the app, and how they might be useful.

counting people during lockdown is an excellent example to use this app. Also check the time spent in markets is a good commercial explanation.

Well done!

Discuss lighting, model accuracy, and camera focal length/image size, and the effects these may have on an end user requirement.

The impact of light on the accuracy of the model explained neatly.

If a suitably accurate model is not found, and the student instead utilizes an existing Intermediate Representations provided by Intel®, *at least three* model attempts should be documented as follows:

- Where the model was obtained from
- How to convert the model to an IR
- Why the model failed and attempts to get it to succeed, such as adjusting confidence threshold

DOWNLOAD PROJECT

RETURN TO PATH

Rate this project