



[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

Dear student,

Excellent implementation of the app. It shows the clarity in your understanding the hard work put in. All the best with the next projects

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).

Excellent job selecting the `ssd_mobilenet_v2_coco`, converting it and running it. Works well!

✓
In `inference.py`, a `load_model()` function is implemented to load the model into the Inference Engine.

نشر تطبيق Counter People على الحافة
translated from: English

Good job implementing the `load_model()` function to load the model into the inference engine

✓
`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.

Good job checking if the layers and CPU extension support

✓
Requests for inference are handled asynchronously through the Inference Engine.

Correctly implemented asynchronous capability using `async_infer()` function

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

Good job extracting the output from the inference response

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.

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

Good job resizing the frame to the model's input specifications

✓
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.

The bounding boxes are drawn correctly and the image is mapped back to the original size

Statistics regarding نشر تطبيق Counter People على الحافة on screen, and total people counted are calculated based of translated from: English antic masks.

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

Statistics regarding people count on screen, duration and total count is calculated and displayed correctly

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.

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.

Well explained the what custom layers are and how to convert them

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.

You have discussed the performance in terms of model size and inference time correctly for pre and post model conversion

نشر تطبيق Counter People على الحافة
translated from: English

Explain potential use

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

Good job listing and explaining the potential use cases of the app



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



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

Your selected model performs well and hence you didn't need to use an Intel provided IR

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)