

Yolo v5 Model Card

Model Card

Model Details

- The YOLOv5s-416 model is a quantized version of YOLOv5s, optimized for efficient object detection. YOLOv5 is a single-stage object detection model that processes an image in one pass through the neural network to detect objects, making it faster than two-stage detectors like Faster R-CNN. The “s” variant (small) is a lightweight version designed for lower latency and efficiency. The INT8 quantization reduces model size and computational complexity, enabling faster inference while slightly trading off precision. The model takes a 416x416 RGB image as input and applies convolutional layers and anchor-based detection to identify objects within the image. It outputs bounding boxes, class labels, and confidence scores, indicating the presence and location of objects.

Intended Use

- Our application uses this model for AI inferencing on input video and we collect metrics while the pipeline is running
- The quantized version is optimized for Intel hardware using OpenVINO’s Inference Engine, making it suitable for applications requiring real-time detection, such as video analytics, autonomous navigation, and smart surveillance.

Training and validation data

- We are not training or validating this model in our reference implementation

Ethical Considerations

- We are using person-bicycle-car-detection.mp4 from <https://github.com/intel-iot-devkit/sample-videos> as input video to test this application tool.
- We are not storing any person or user related personal information.

Caveats and Considerations

- The model's accuracy may vary depending on the quality and resolution of the input images. Ensure that the images used are of sufficient quality for reliable detection.
- Preprocess images to normalize lighting conditions and remove noise.

Quantitative Analysis

- We are not doing quantitative analysis in this application tool but we do display metrics mentioned below to the user.

Factors

- We are also not evaluating this model in this reference implementation

Metrics

- We are displaying metrics including throughout (FPS) and system level metrics: CPU/GPU utilization, memory utilization, CPU/GPU frequency, CPU/system temp, GPU power, GPU engine, and package power. In this application these metrics are collected and displayed to the user via gauges.