Deploying Real Time Object Detection Inference Systems

Yves Kreimerman, 2025

Abstract. Computer vision is a field in machine learning that aims to understand a give picture, identify what's going on and classify those into classes(patterns) it understands. There are couple of main fields in the computer vision world, those being:

1. **Classification** - The model classifies a whole picture into classes. It is possible for the model to classify the picture as multiple classes(Multi-Label Classification), or one class, being the most dominant(Multi-Class Classification)

2. **Object Detection** - The model omits a series of boxes around various object it identifies(based on the data it was trained), pointing to the coordinates where the object is found on the picture.

3. **Semantic Segmentation** - The model identifies pixels as part of a class, but does not distinct between different objects sharing the same class

4. **Instance Segmentation**- The model identifies pixels as part of a class, and has the distinction between objects. it is able to distinct two objects from the same class that are nearby.

1. Introduction

Live streaming video is a broad subject, of producing a live video feed which is constantly broadcasted from various sources, with consumers watching on the other end. Live streaming is usually split into couple of main categories, the main ones being:

Type	Delay Range (Seconds)	Use
WebRTC	0.5 - 2	Real Time (video calls, surveillance)
Broadcast Live	2 - 5	TV (News, Sports)
RTMP/HLS	5 - 30+	Live Media (Youtube, Twitch)

Table 1: Main categories of live streaming

Having Machine Learning fit model into the chain of live streaming can be difficult, as it requires us to fit under very slim time constraints.
With Real Time being our main subject, we will go over the constraints we should take into count, and how they project on our system design:

1. **Source Capture & Extraction of raw frame** - Getting raw frames from source, before any manipulations - <u>30ms</u>

2. **Machine Learning Inference** - Processing, model inference, postprocessing

3. **Video encoding & Processing** - Processing frames into actual video, encoding to various formats(e.g. H264) - <u>20ms</u>

4. **Network Transport** - Transporting the processed video to consumer(Over LAN/WAN) - <u>100ms-150ms</u>

5. **Consumer Decoding & Display** - Displaying the video to the end consumer via streaming platform - <u>100ms-130ms</u>

The numbers shown are for an ideal scenario, and in reality numbers are much higher. That requires us to fit under very low constraints, ideally around **15ms-75ms**, depending on video frame rate and network throughput.

3. Methodology

Having tight time constraints to take into consideration, we would have to design a low-latency, high throughput system, that can serve as many video sources as possible, while utilizing the most from the given resources.

3.1 Model Precision

With the goal of minimizing latency of inferance, and maximizing processing throughput, we need a way to optimize the given ML models and squeeze the best preformance out of those.
By default, machine learning models use FP32(Floating point 32) datatypes for weights and inputs/outputs by default, allowing higher precision for training, therefore maximizing accuracy. When running inference on the same models, we don't necessarily need all that precision and can "compromise" on lower floating points, i.e. FP16. FP16 should give us much better preformance, while almost not compromising on accuracy.
We first want to test all model versions for accuracy, determine how lossy is the conversion of floating points and is compromising on floating points even worth it.
We will do so with a pre-made evaluation script, while models are evaluated on MS-COCO Val 2017 dataset, for consistency purposes(all models are trained on the same data too).
The goal is to demonstrate how one model compares to another, so the actual performance of the model(how well it performs on the data) is trivial.
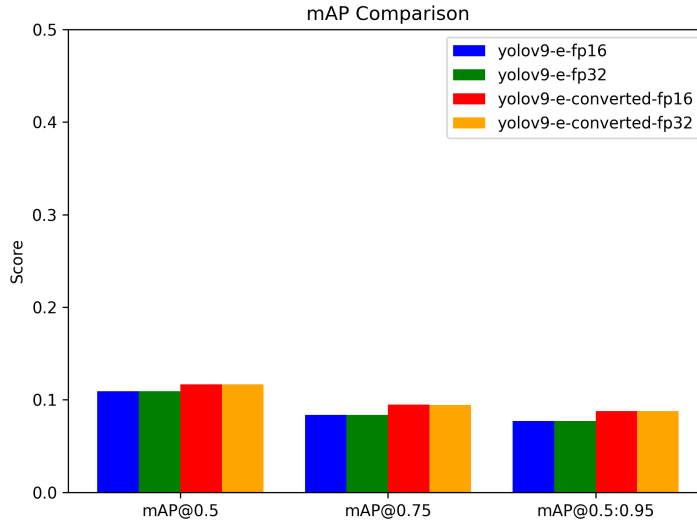

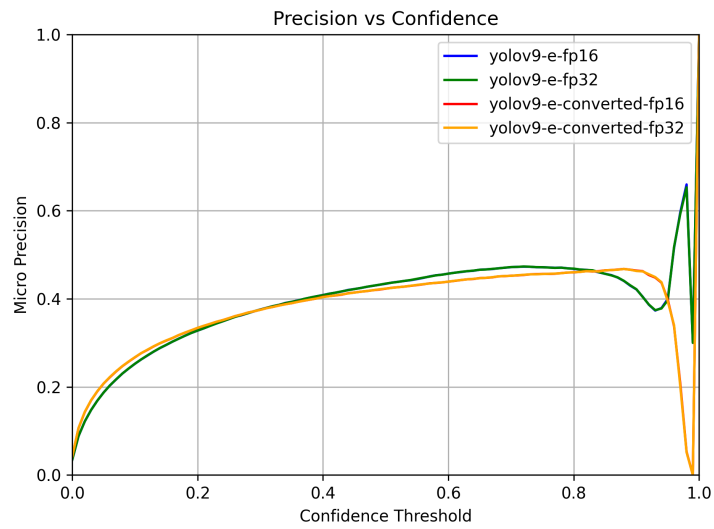Figure 1: mAP (under different IOUs)

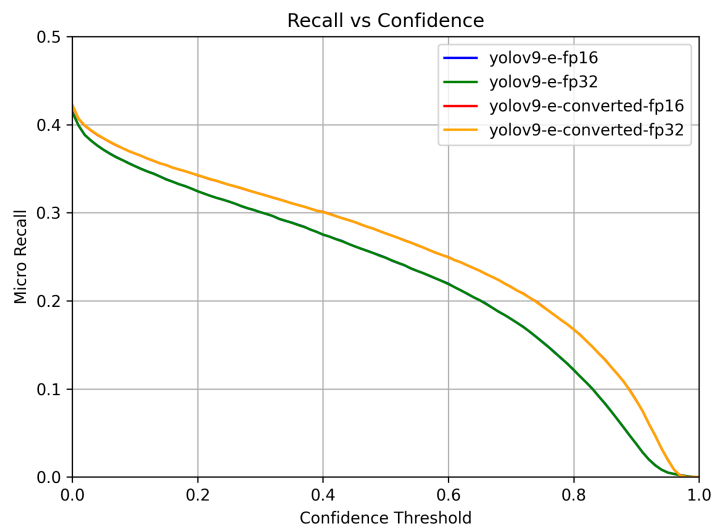Figure 2: Precision - Confidence Curve
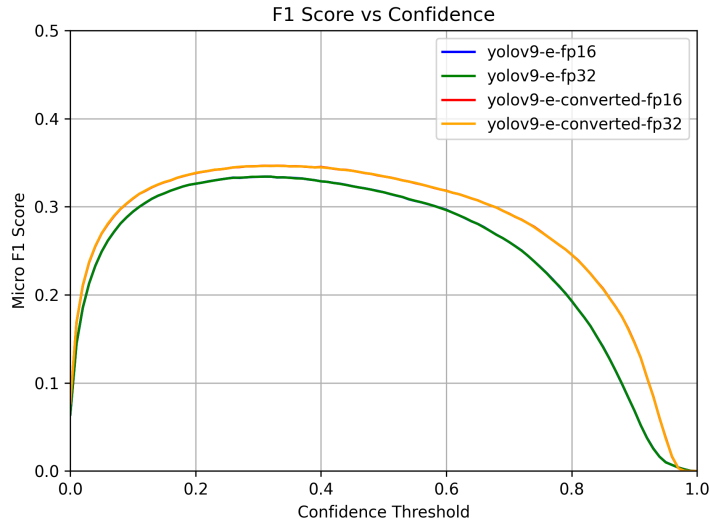

Figure 3: Recall - Confidence Curve

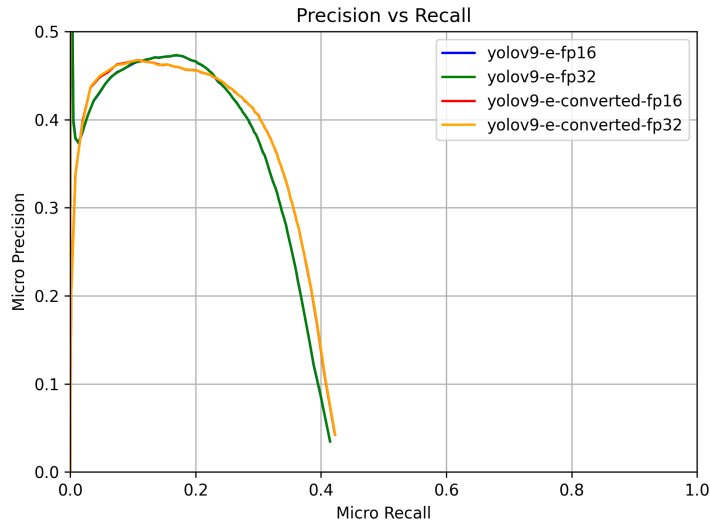Figure 4: F1 Score - Confidence Curve

Figure 5: Precision - Recall Curve

While it may not be visible right away, we can see that FP16 and FP32 results for the same kind of model are overlapping in all curves. This means that they both reach identical results(with minor changes after the 4th floating digit).

With that said, **FP16 models will be our choice of deployment** as they allow us to have the same inference accuracy while significantly lowering inference time(higher throughput).