A comparative study on One-stage vs. two-stage object detectors

One-stage object detectors refer to neural networks anticipating all the bounding boxes in a single trip through the web. Mobile devices are better suited for these tasks due to increased speed and compatibility. Some of the most often encountered instances of one-stage object detectors include YOLO, SSD, SqueezeDet, and DetectNet.In contrast, two-stage object detectors employ a two-step approach. Initially, they utilize region suggestions to create preliminary object proposals. Subsequently, a specialized per-region head categorizes and enhances these ideas. One-stage detectors typically exhibit faster processing speeds, while two-stage detectors generally provide superior accuracy.

The prevalent architectures for one-stage object detection include YOLO, SSD, and EfficientDet. The YOLO (You Only Look Once) system is an object recognition system that operates in real time. It utilizes a singular neural network to make predictions regarding bounding boxes and class probabilities straight from whole images in a single evaluation. Single Shot Detection (SSD) is a one-stage object detection method that uses a solitary deep neural network to immediately estimate bounding boxes and class probabilities from complete images in a single evaluation. EfficientDet refers to a collection of one-stage object detectors that employ EfficientNet as the underlying network architecture, demonstrating exceptional performance on diverse object detection benchmarks.

One-stage object detectors offer several advantages, including enhanced speed and excellent compatibility with mobile devices. In addition, they necessitate a reduced amount of memory and computational resources. One of the drawbacks of single-stage object detectors is their relatively lower accuracy than two-stage detectors. Additionally, these detectors may have challenges in accurately identifying small items or objects close to each other.

One advantage of two-stage object detectors is their tendency to achieve higher accuracy levels than one-stage detectors. Additionally, they exhibit enhanced proficiency in identifying things with irregular shapes or clusters of smaller objects. One of the drawbacks of two-stage detectors is their slower processing speed and increased need for memory and computational resources compared to one-stage sensors.

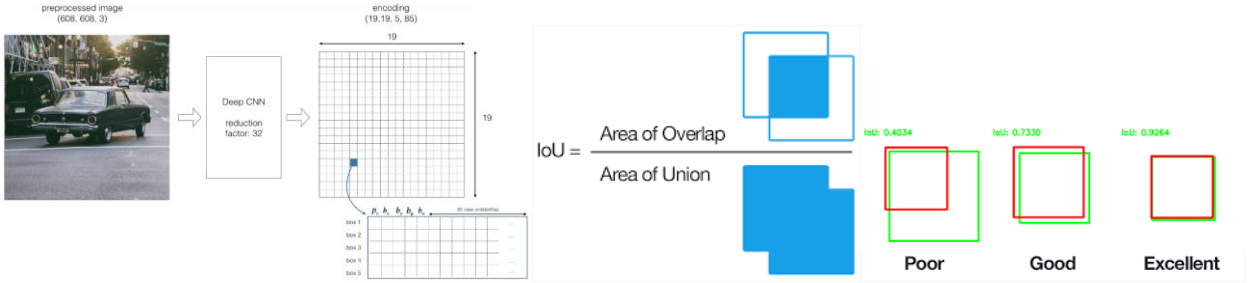
Yolo:

- In YOLO, Object detection as a single regression problem also known as single stage detector

  bbox coordinates & class probabilities all are computed in just a single run of algorithm

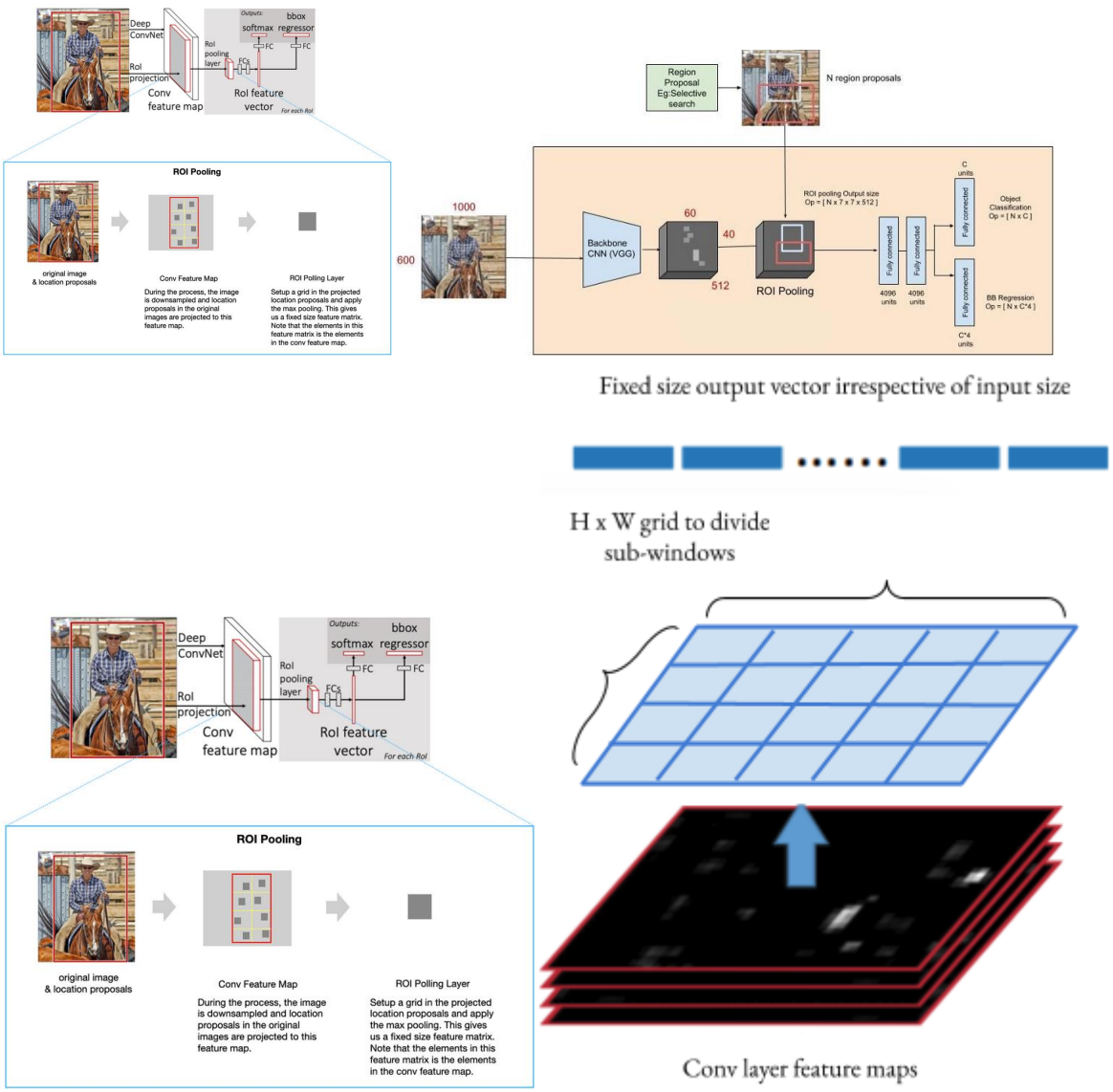
- YOLO sees the entire image during training & test time so it makes < 1/2 the no. of background errors compared to Fast R-CNN.
- Intersection Over Union (IOU) is a metric used to judge the accuracy of the bounding box predicted by the model.
- Non-max Suppression reduces the no. of predicted b.boxes by taking the larges probability associated with each detection
- **Drawbacks:**

- The major problem with YOLOv1 is its inability to detect very small objects.



Fast R-CNN:

- Similar to R-CNN, just that it expects region proposals to be fed in with images rather than proposing them itself
- We run the CNN only once per image sharing the computation among 2,000 region proposals.
- The CNN processes the image and outputs a feature map
- Input region proposals are used to extract the ROI from feature map & create a region proposal feature map for each proposed region called a the ROI Projection
- Then down-sample this feature map with the help of a ROI pooling layer to get a fixed length feature map
- **Drawbacks:**
- much faster as compared to R-CNN. But with large real-life datasets, it does not works so fast



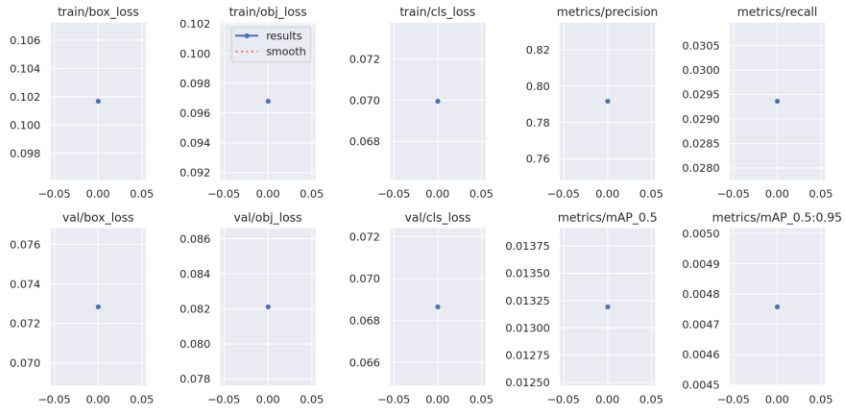
Result:

Here is Fast R-CNN result

```
] scores

tensor([0.9943, 0.9886, 0.9625, 0.9046, 0.9044, 0.9027, 0.8800, 0.8290, 0.8278,
        0.8212, 0.7828, 0.7669, 0.6758, 0.6336, 0.5739, 0.5215, 0.5204, 0.4837,
        0.4741, 0.4501, 0.4474, 0.4294, 0.4084, 0.4043, 0.3769, 0.3628, 0.3377,
        0.3350, 0.3132, 0.2829, 0.2650, 0.2508, 0.2496, 0.2413, 0.2190, 0.2135,
        0.1736, 0.1730, 0.1327, 0.1309, 0.1267, 0.1263, 0.1160, 0.1145, 0.1102,
        0.1016, 0.0982, 0.0886, 0.0869, 0.0858, 0.0808, 0.0737, 0.0728, 0.0678,
        0.0673, 0.0658, 0.0650, 0.0622, 0.0616, 0.0575, 0.0547, 0.0534, 0.0515,
        0.0501])
```

Here is Yolov5 result



In general, Faster R-CNN is a good choice for applications where accuracy is critical. YOLOv5 is a good choice for applications where speed is critical.

Feature	Faster R-CNN	YOLOv5
Accuracy	High	Medium
Speed	Slow	Fast
Memory usage	High	Medium
Complexity	High	Medium

Ultimately, the best model for a particular application will depend on the specific requirements of that application.

In some cases, Faster R-CNN may be the better choice, even if speed is a concern. For example, if the application is only processing a small number of images, the slower speed of Faster R-CNN may not be a significant drawback.

In other cases, YOLOv5 may be the better choice, even if accuracy is a concern. For example, if the application is processing a large number of images in real time, the faster speed of YOLOv5 may be essential.

The best way to choose between Faster R-CNN and YOLOv5 is to experiment with both models and see which one works best for the particular application.

Reference:

1. *https://www.analyticsvidhya.com/blog/2022/06/yolo-algorithm-for-custom-object-detection/*

2. *https://viso.ai/deep-learning/object-detection/#:~:text=on%20Viso%20Suite-,Most%20Popular%20Object%20Detection%20Algorithms,the%20single%2Dshot%20detector%20family*
3. *https://pjreddie.com/darknet/yolo/*
4. *https://www.v7labs.com/blog/yolo-object-detection*