YOLO Object Detection Write Up

Conclusion: “Currently, Faster-RCNN is the choice if you are fanatic about the accuracy numbers. However, if you are strapped for computation(probably running it on Nvidia Jetsons), SSD is a better recommendation. Finally, if accuracy is not too much of a concern but you want to go super fast, YOLO will be the way to go.”

**My Thoughts**: I think it’s worth to try all three different computational algorithms, but if we are concerned with accuracy especially since we are not too worried about object detection runtime as we aren’t doing real-time detection like in autonomous driving, it may be in our best interest to look into RCNN or SSD algorithms.

YOLO Darknet Object Detection Algorithm

1. About YOLO Darknet (https://pjreddie.com/darknet/yolo/)
   1. The original creator of YOLO Darknet is responsible only for versions until YOLOv3, with YOLOv4 and YOLOv5 created by other users who looked to build upon the old YOLOv3 Algorithm. Just last year, YOLOv7 was released and provides better accuracy than older versions.
2. Pros of YOLO Darknet
   1. Ease of Use
      1. The training of R-CNN (Recurrent Convolutional Neural Networks) is often time consuming and cumbersome to use in practice.
         1. The use of recursion means that these models take extremely long times to train and also take longer times to detect objects (more on this below).
      2. YOLO Darknet is extremely easy to set up and train weights. Importing custom datasets is also relatively straightforward by using LabelImg to label data before feeding it into the algorithm for training.
   2. Low Cost
   3. Good Object Detection Accuracy and Speed
      1. YOLOv7 maintains the high processing speed that has become synonymous with YOLO over the years, processing images at speeds of up to 155 frames per second.
      2. YOLOv7 also has good prediction accuracy, performing similarly to other state-of-the-art object detection algorithms. However, YOLOv7 is outperformed in accuracy by certain R-CNN, but R-CNN’s often take much longer to generate predictions and inferences.
3. Cons of YOLO Darknet
   1. Detection Accuracy is a tradeoff for Detection Speed/Does Detection Speed Matter for our Project?
      1. In this [paper](https://assets.researchsquare.com/files/rs-668895/v1_covered.pdf?c=1631875157), YOLO Darknet is outperformed by Faster R-CNN in terms of Mean Accuracy Precision (MAP), but YOLO Darknet has a much faster object detection speed. However, does object detection speed matter for our project? Would it be better to sacrifice detection speed for detection accuracy, given that we are not applying our object detection algorithm in real time, but rather on still images?
   2. Small Object Detection
      1. While not limited to just YOLOv7 as many other object detection algorithms run into the same issue, YOLO Darknet sometimes has issues detecting smaller objects. Whether or not this can be offset by having higher quantities of smaller items is something that we will have to experiment with and figure out.
   3. Transfer Training/Pre-Trained Weights
      1. While YOLO has no pre-trained weights, the usage of R-CNN or other neural networks allows us to use pre-trained weights or transfer training, where we can train pre-tuned weights from other models on our model. Since the pre-trained weights already have patterns and inferences, they may enable R-CNNs to perform better on smaller datasets compared to YOLO Darknet.

Alternatives

**R-CNN (Recurrent Convolutional Neural Networks)**

1. Pros of R-CNN
   1. The use of Faster/Mask R-CNN enables the use of pre-trained weights while also keeping training time to a minimum. The recursive aspect of R-CNN means that training times will be much greater than that for the YOLO Darknet, but studies have also seen higher accuracy precision when using R-CNN for object detection.
   2. The use of Faster R-CNN also enables us to train models with the same accuracy and precision while using less data inputs, something that may prove advantageous given some of the data issues we may run into.
   3. The reason why YOLO Darknet is so popular is due to its fast object detection time as stated above, but given our project, is fast object detection necessary? Is it possible to trade off object detection speed (since we’re not doing in real time detection) for better mean accuracy precision in object detection predictions?
2. Cons of R-CNN
   1. Long training and identification times. However, long training times can be mitigated by using the Faster R-CNN which decreases the training time significantly, and identification times take up to one minute to identify. Will this one minute take too long to use, especially if we focus on ease of use? The user may expect to send a picture and have instant feedback instead of having to wait for results.

**Single Shot Detector (SSD)**

1. Pros of SSD
   1. Compared to YOLO, SSD has better detection accuracy but a longer object detection time. SSD does not suffer from the same issues that YOLO suffers from when images are extremely close. Rather, SSD has better coverage on location, scale, and aspect ratios.
   2. Achieves a balance between speed and accuracy. SSD only passes through the image once similarly to YOLO, but uses a Deep Convolutional Network to calculate feature maps.
2. Cons of SSD
   1. SSD is known to suffer more than YOLO when dealing with objects of smaller sizes. While all object-detection algorithms tend to struggle with identifying smaller objects, SSD is an algorithm that struggles especially.

Sources

<https://www.v7labs.com/blog/yolo-object-detection#:~:text=One%20of%20the%20main%20advantages,driving%20cars%2C%20and%20augmented%20reality>.

<https://blog.paperspace.com/yolov7/#how-does-yolo-work>

<https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>

<https://mobidev.biz/blog/object-detection-small-datasets-use-cases-machine-learning>

<https://viso.ai/deep-learning/object-detection/#:~:text=on%20Viso%20Suite-,Most%20Popular%20Object%20Detection%20Algorithms,the%20single%2Dshot%20detector%20family>.

<https://algoscale.com/blog/yolo-vs-ssd-which-one-is-a-superior-algorithm/#:~:text=YOLO%20>(You%20Only%20Look%20Once,forward%20pass%20computing%20feature%20map.

<https://medium.com/ibm-data-ai/faster-r-cnn-vs-yolo-vs-ssd-object-detection-algorithms-18badb0e02dc>

<https://blog.paperspace.com/mask-r-cnn-in-tensorflow-2-0/>