IE534 Final Project Proposal: Faster R-CNN

Group 16
Sanyukta Deshpande(spd4), Refik Mert Cam(rcam2)
Abhinav Garg(garg19)

March 20, 2021

1 Introduction

As the final project of the IE534-Deep Learning course, we have chosen to implement Faster R-CNN architecture [1]. Faster R-CNN is a state-of-the-art member of the R-CNN family; which is developed for object detection tasks. Its predecessors are Regions with CNN (R-CNN) and Fast R-CNN. The evolution between versions is motivated by the aim to increase computational efficiency and decrease the test time. The R-CNN family consists of a region proposal algorithm to extract the regions in an image where possible objects locate, a feature generation layer to extract the features of the objects, a classification layer to label the objects, and a fine-tuning layer to make the localization of the object more precise. The predecessors; R-CNN and Fast R-CNN, use CPU-based region proposal methods, which is the selective search algorithm that takes around 2 seconds per image. The Faster R-CNN replaces the selective search algorithm with a Regional Proposal Network (RPN), which uses CNN, and increases the computational efficiency. The RPN component provides "attention" and tells the unified network where to look. Furthermore, as the RPN shares its convolutional features with the object detection network, this innovation is almost cost-free.

2 Specifications

Our aim is to implement the Faster R-CNN with the following specifications and reproduce the results in [1].

2.1 Backbone network selection

As the backbone CNN architecture; which is shared by both object detection network and region proposal network, we plan to use VGG and ResNet architectures as two different cases.

2.2 Dataset selection

The original paper uses PASCAL VOC 2007, 2012 and COCO datasets. We plan to use PASCAL VOC 2007 dataset; which contains 9963 images and 24640 annotated objects in total, because the time is a restriction for this project. If time permits, we may explore the performance over different interesting datasets.

2.3 Training strategy

The original paper proposes three different training strategy: (i) alternating training, (ii) approximate joint training, and (iii) non-approximate joint training. We plan to explore (i), and (ii) strategies in our implementation.

2.4 Scale and aspect ratio of Anchors

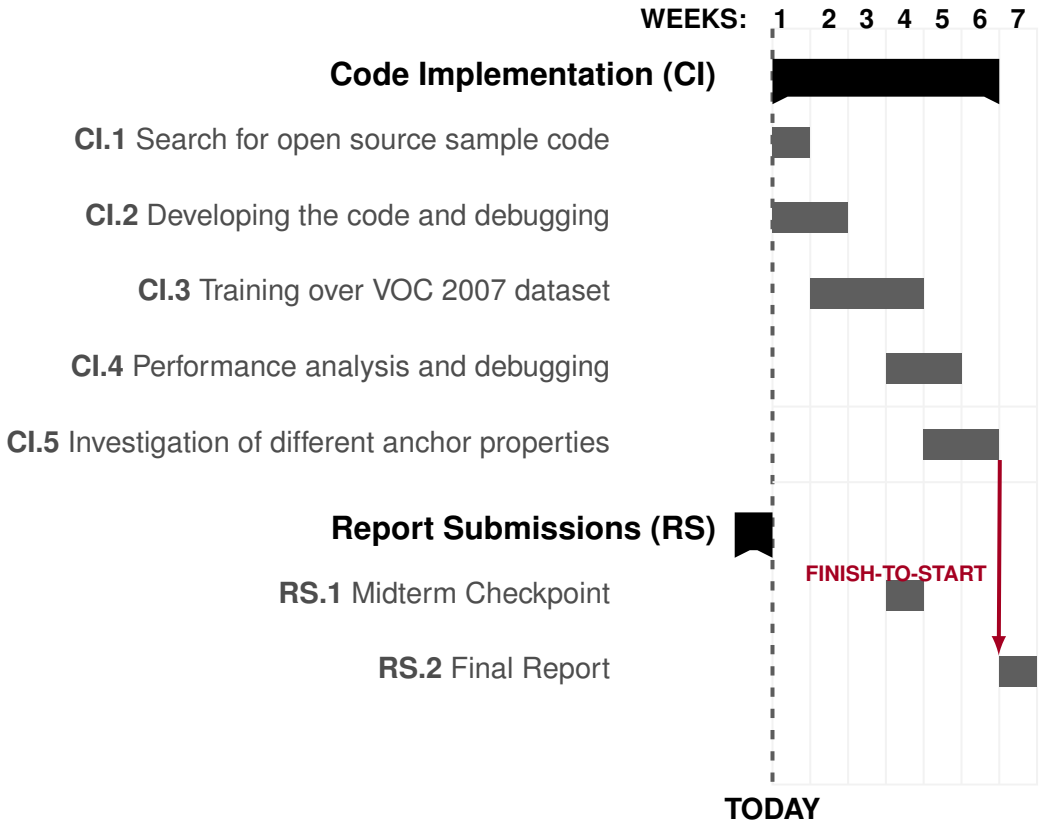
In the original paper, they use 3 scales and 3 aspect ratios. We plan to explore all of them in our implementation.

3 Deliverables

In the final report of this project, we aim to provide the following performance measures, using the above specifications:

- Loss curves
- mAP (mean average precision) both for different classes and over the general dataset
- Sample outcomes

4 Estimate timeline and task division of the project



In [2], it is indicated that training of Fast R-CNN takes 8.75 gpu hours. As we will conduct approximately 12-14 training (including hyperparameter tuning), we expect that whole project will take 105-122.5 gpu hours.

References

[1] Faster RCNN Implement “Faster R-CNN: Towards real-time object detection with regional proposal networks”, NIPS, 2015 by Ren et al.

[2] R. Gandhi, “R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms,” Medium, 09-Jul-2018. [Online]. Available: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e. [Accessed: 20-Mar-2021].