To Showcase custom Object Detection on the Given Dataset to train and Infer the Model

EagleView Task Assignment for Custom Training with YOLOv5

Contact Information

- **Name** - Owais Ahmad
- **Phone** - +91-9515884381
- **Email** - owaiskhan9654@gmail.com
- **Portfolio** - https://owaiskhan9654.github.io/

Current Designation

Data Scientist at Canary Global Inc, Bangalore, India.

Education

Masters of Technology in Statistical Computing,

School of Computer and System Sciences, Jawaharlal Nehru University,

New Delhi, India

Code link

I have uploaded all the code and work on my GitHub Repository. You may access it from here.
https://github.com/Owaiskhan9654/EagleView-Custom-Object-Detection

Google Colab Notebook link -
https://colab.research.google.com/drive/1Nk1oDT4WqgVOOgXSjMSVLjVc-4JTzDgx?usp=sharing

High Level Documentation

Introduction

With the evolution of digital communication technology and innovative Deep learning tasks, the ability to offer high-quality support Image processing tasks has stemmed magnificent new capabilities. The success is subjected to understanding the intrinsic processes being used in the Image processing pathway by which normal users can make decisions faster.

Methods

- To detect & classify the object in the image using YoloV5 custom Object Detection Model.
- This write-up gives a Flowchart of how it is working with step-by-step methods.

ROADMAP

1. Process the given Dataset with RoboFlow. In given dataset Images of Different Dimensions were available and also data was not split into train test split.
2. Export the dataset to YOLOv5
3. Train YOLOv5 to recognize the objects in our dataset
4. Evaluate our YOLOv5 model's performance
5. Run test inference to view the performance of the custom YOLOv5 model at work

EagleView Custom Training with YOLOv5

Object detection is a task in computer vision that involves identifying the presence, location, and type of one or more objects in a given photograph.

It is a challenging problem that involves building upon methods for object recognition (e.g. where are they), object localization (e.g. what are their extent), and object classification (e.g. what are they).

In recent years, deep learning techniques are achieving state-of-the-art results for object detection, such as on standard benchmark datasets and in computer vision competitions. Notable is the “You Only Look Once,” or YOLO, a family of Convolutional Neural Networks that achieve near state-of-the-art results with a single end-to-end model that can perform object detection in real-time.

There are five main variations of the approach, at the time of writing; they are YOLOv1, YOLOv2, YOLOv3, YOLOv4, and YOLOv5. The first version proposed the general architecture, whereas the second version refined the design and made use of predefined anchor boxes to improve the bounding box proposal, and version three further refined the model architecture and training process.

YOLOv5 is extremely fast and accurate
The official ultralytics GitHub repository contains the source code for the YOLO versions mentioned in the papers, written in Pytorch. The repository provides a step-by-step tutorial on how to use the code for object detection.
I used COCO class labels, used a pre-trained weight and yolov5 configuration on which YOLO model was trained for this custom object detection model

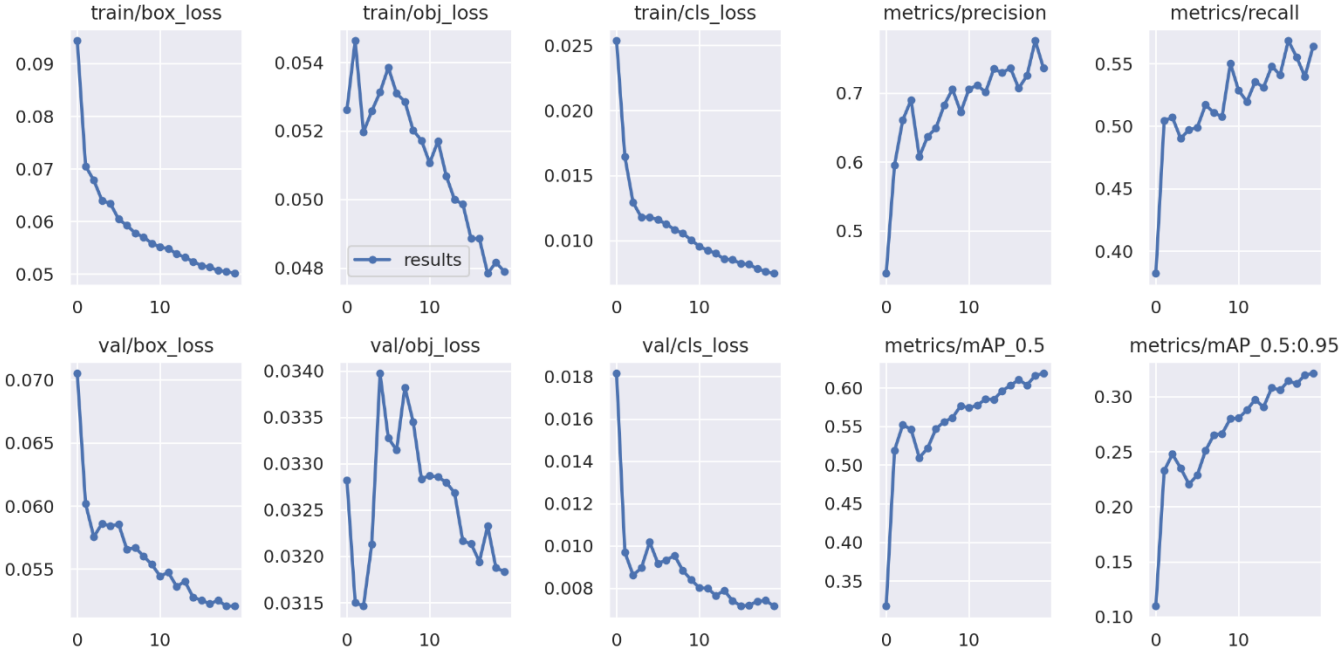

Fig 1.1 Training and Validation Loss and mAP

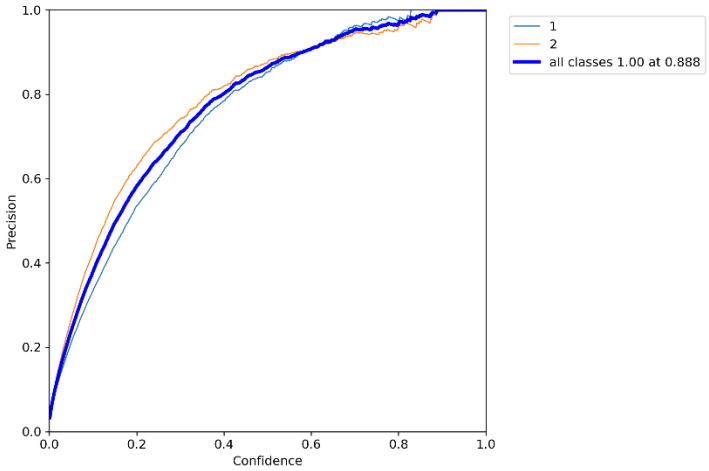

Fig 1.2 Precision vs Confidence

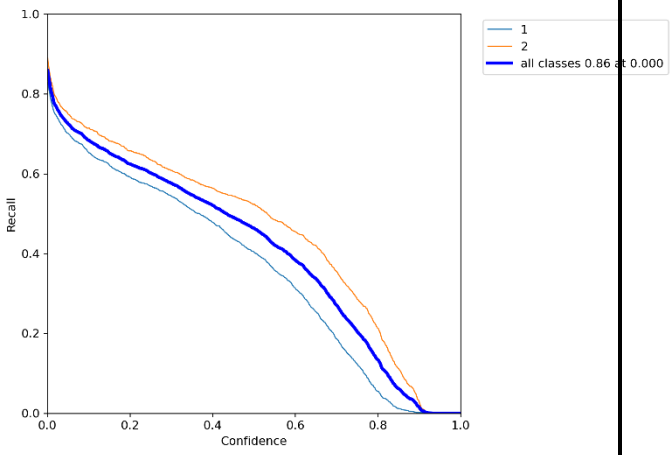

Fig 1.3 Recall vs Confidence

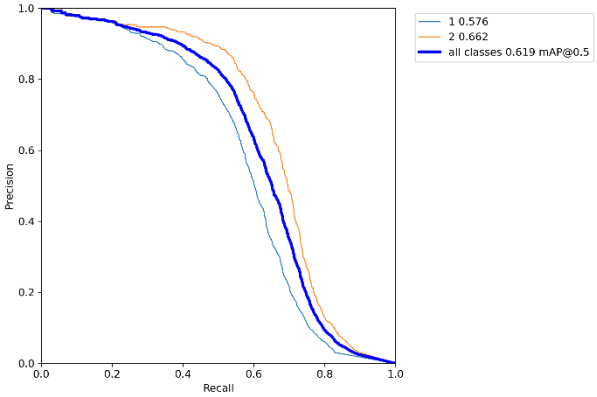

Fig 1.4 Precision vs Recall

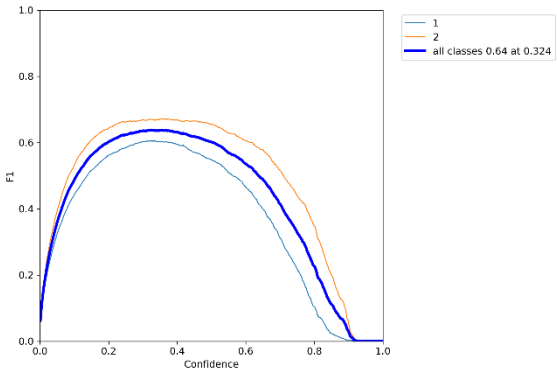

Fig 1.5 Precision vs Recall

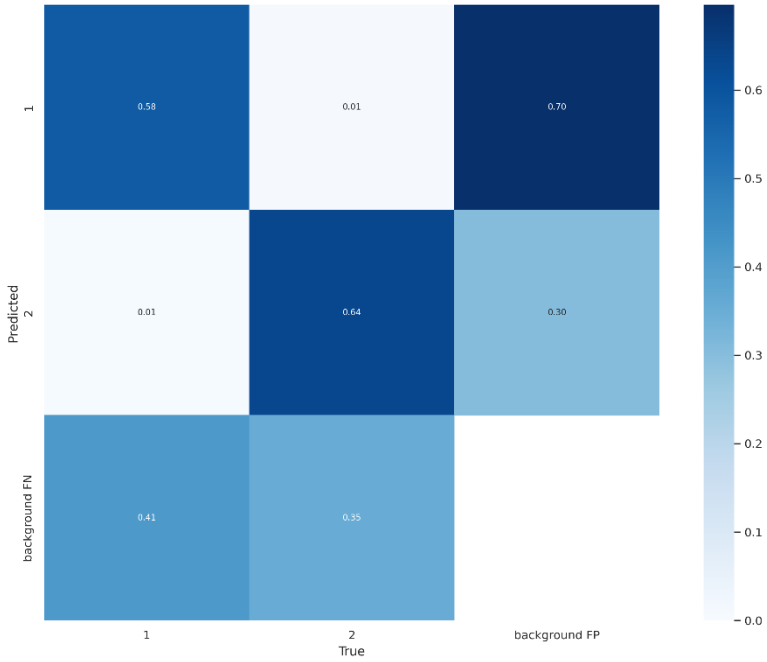

Fig 1.6 Confusion Matrix

epoch	train/box_loss	train/obj_loss	train/cls_loss	metrics/precision	train/box_loss	train/obj_loss
0	0.094373	0.052611	0.025357	0.43813	0.094373	0.052611
1	0.070509	0.054635	0.016439	0.5953	0.070509	0.054635
2	0.067797	0.051962	0.012916	0.66051	0.067797	0.051962
3	0.063923	0.052579	0.011782	0.68976	0.063923	0.052579
4	0.06338	0.053129	0.011809	0.60718	0.06338	0.053129

Some results on Training Epochs Metrics

References

1. https://github.com/ultralytics/yolov5/
2. https://ultralytics.com/
3. https://roboflow.com/
4. https://www.murtazahassan.com/courses/opencv-projects/lesson/lesson-5/
5. https://www.youtube.com/watch?v=iIOYQOfUi8U&ab_channel=Life2Coding
6. https://pjreddie.com/darknet/yolo/
7. https://medium.com/analytics-vidhya/object-detection-using-yolo-v3-and-deploying-it-on-docker-and-minikube-c1192e81ae7a

Code link

I have uploaded all the code and work on my GitHub Repository. You may access it from here.
https://github.com/Owaiskhan9654/EagleView-Custom-Object-Detection

Google Colab Notebook link -

https://colab.research.google.com/drive/1Nk1oDT4WqgVOOgXSjMSVLjVc-4JTzDgx?usp=sharing