Detection of person and cars

By Madhuri Konnur

Abstract

There are many image data as well as models are available to detect objects. Majority of them deal with output which contains bounding box coordinates, class scores and confidence values etc.
Here I am planning to utilize Fiftyone app for the rescoring approach of object detection.
[Contextual rescoring takes detections as input and recomputes confidence scores to improve the mean average precision (mAP) of the detections with the ground truth. Most object detectors produce individual predictions with no context between them resulting in duplicate or out of place predictions.

Tools Used

open-source software : FiftyOne by Voxel51, Google colab notebook to run code.
Libraries : fiftyone, voxel51-eta, eta-install

About FiftyOne App: The open-source tool for building high-quality datasets and computer vision models.FiftyOne supercharges your machine learning workflows by enabling you to visualize datasets and interpret models faster and more effectively.

Dataset

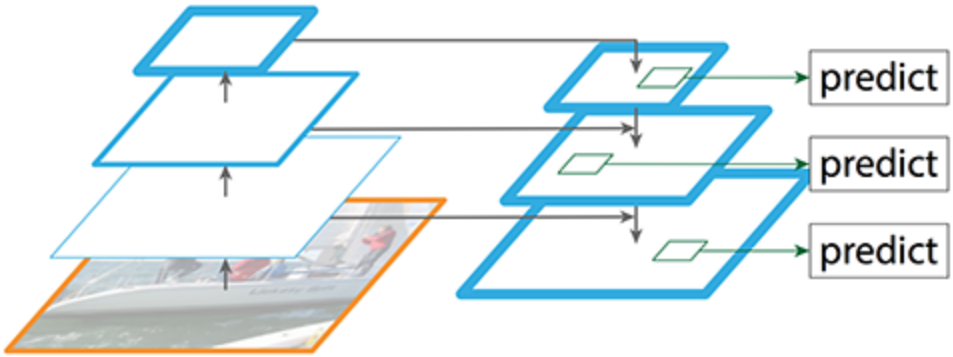
https://evp-ml-data.s3.us-east-2.amazonaws.com/ml-interview/openimages-personcar/trainval.tar.gz

Annotation format:COCO

Model used for detection

FPN Single Shot Detector model from SSD: Single Shot MultiBox Detector-supports detection, coco, tf with MobileNetV1 backbone trained on COCO. FPN-ssd is one of available model from fiftyone zoo library.

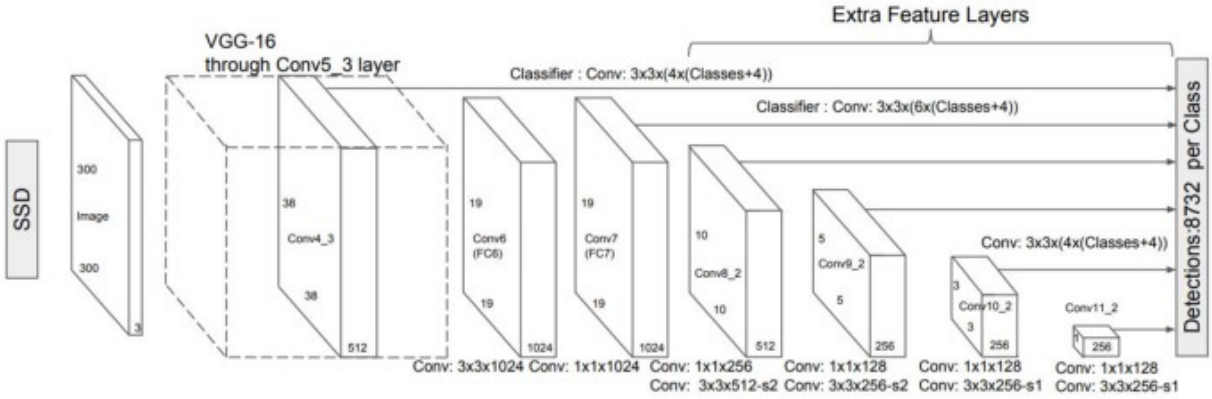
Since, Feature Pyramid Network (FPN) is a feature extractor designed with the feature pyramid concept to improve accuracy and speed. It replaces the feature extractor of detectors like Faster R-CNN and generates higher quality feature map pyramid. FPN composes of a bottom-up and a top-down pathway.



FPN Dataflow

The SSD object detection composes of 2 parts:
Extract feature maps, and Apply convolution filters to detect objects.

SSD uses VGG16 to extract feature maps.Actually, it uses multiple layers (multi-scale feature maps) to detect objects independently.



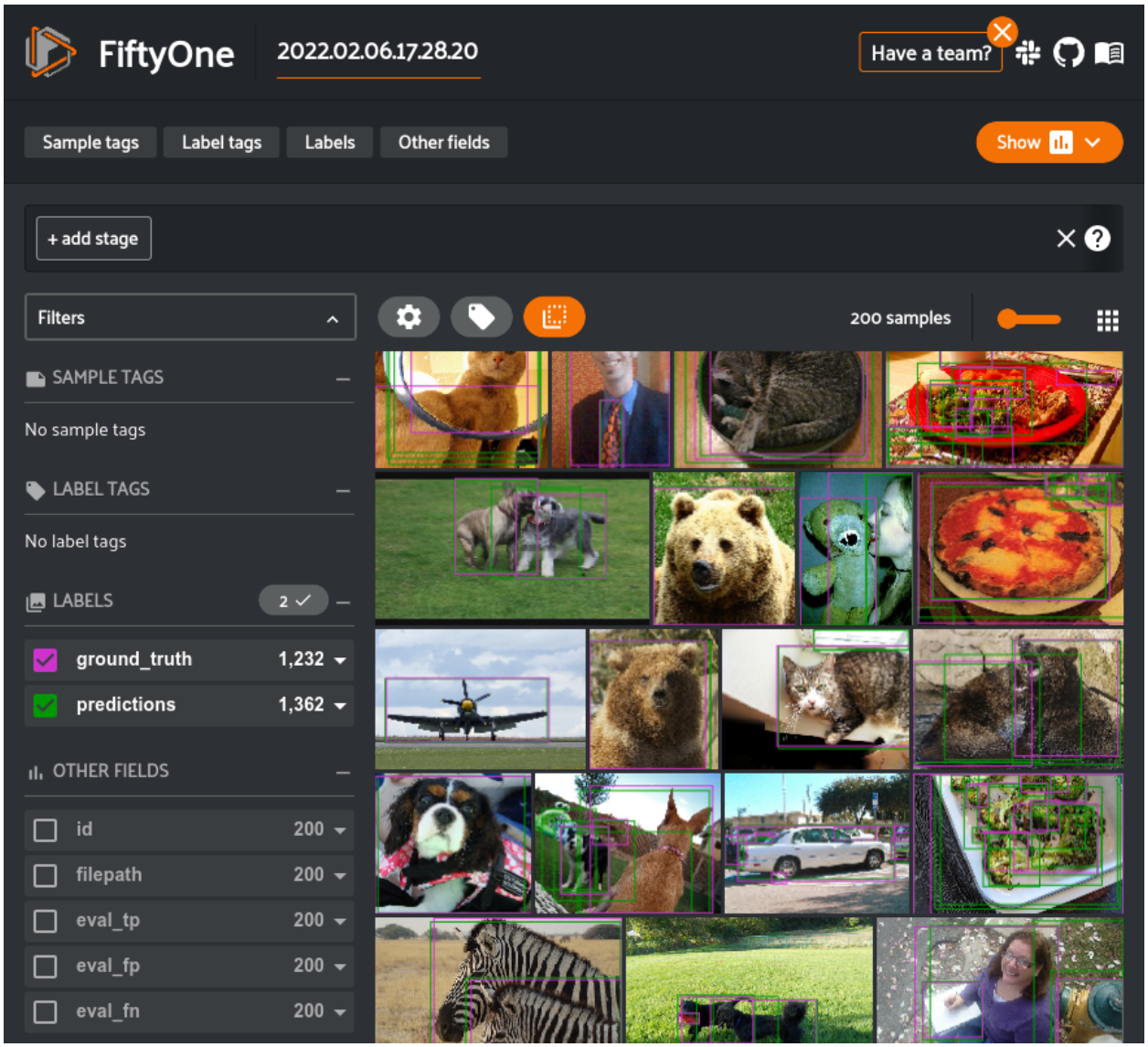
Source: SSD: Single Shot MultiBox Detector

Challenge: Since we have custom data in COCO format .Which was not available on fiftyone zoo library. Hence used add_coco_labels() to conveniently add the labels to an existing dataset.

https://github.com/MadhuriKonnur/Person_Car_Detection/blob/main/Det_mk_EV.ipynb
includes:

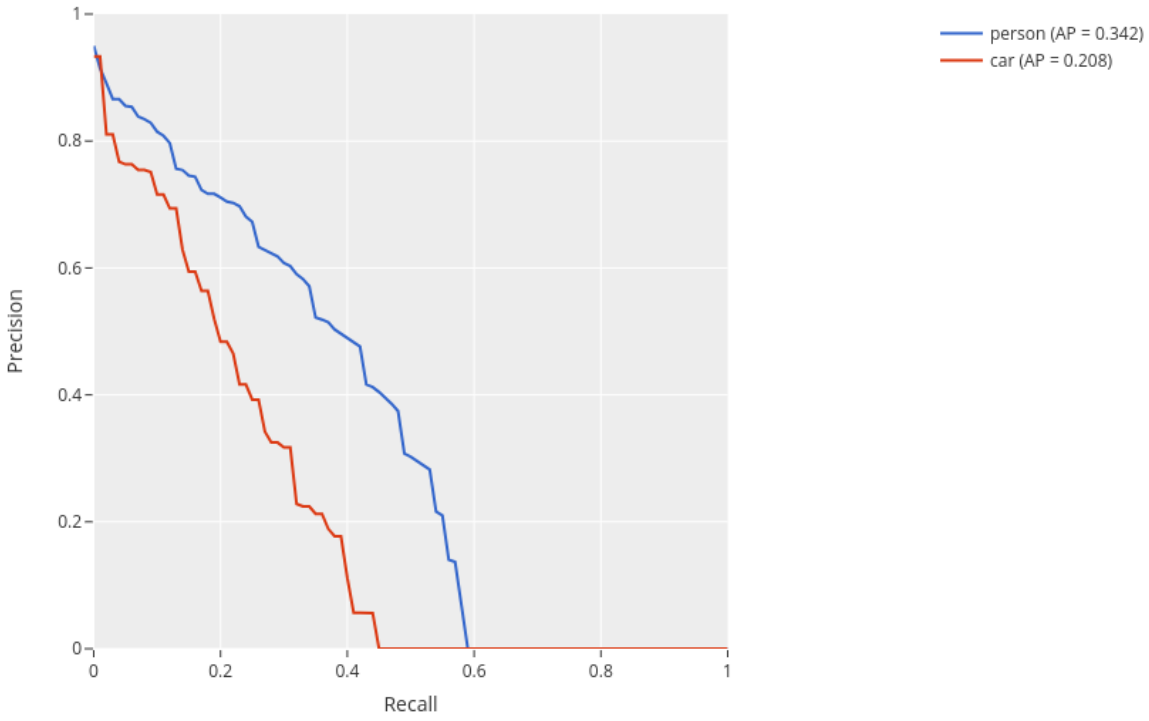
1. round-trip export and then re-import of both images-and-labels and labels-only data in COCO format
2. Model building and training using "ssd-mobilenet-v1-fpn-coco-tf " from available zoo lib

Initial Dataset view:(After loading on quickstart dataset)

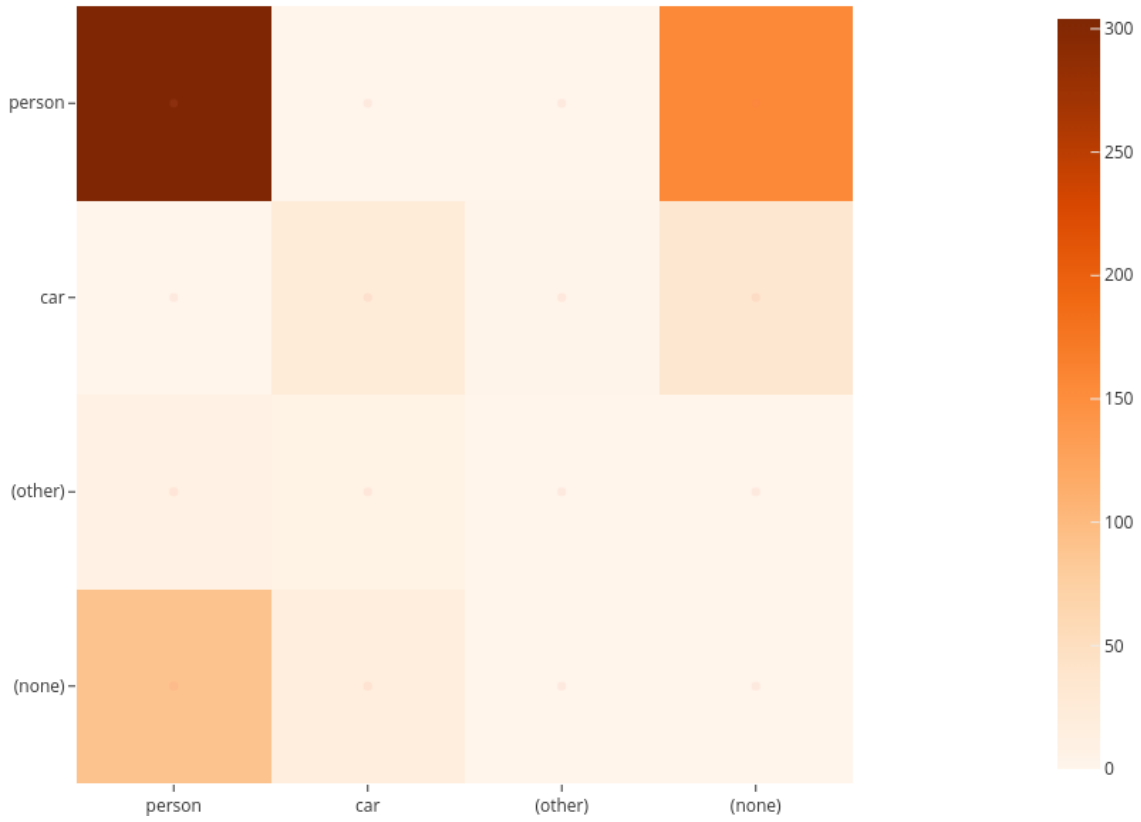


Outcome of the training

mAP:0.367
person AP:0.342
car AP:0.208



Precision and Recall



Confusion Matrix

The slight drop in mAP is expected when matching predicted objects with ground truth of different classes, but this is desirable when trying to closely evaluate and understand your model. By default, when classwise=True, all false positives indicate that a predicted object was left unmatched. On the other hand, with classwise=False, some false positives now indicate that a prediction matched a ground truth object with a different class. This implies that the model was confident about the object being the incorrect class and that is information that we want to know

Recommendations

Need to try loading a custom dataset with other available dataset(Here used "quickstart" dataset).
By tweaking default values like "iou", "classwise" etc.
Analysing errors, patches and evaluating with other detector models.

References

- Pato L., et al, Seeing without Looking: Contextual Rescoring of Object Detections for AP Maximization, CVPR (2020).
- https://voxel51.com/docs/fiftyone/
- https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06