Source Codes Overview

RESNET50 Model

The `rtds-pretrained.ipynb` file is reponsible for fine-tuning the resnet50 model for our application. In this file we have used the popular `fastai` module by Jeremy Howard. This module is basically a wrapper around the pytorch module. So, it abstracts away many setup procedures of pytorch itself.

Why FastAI than pytorch or other?

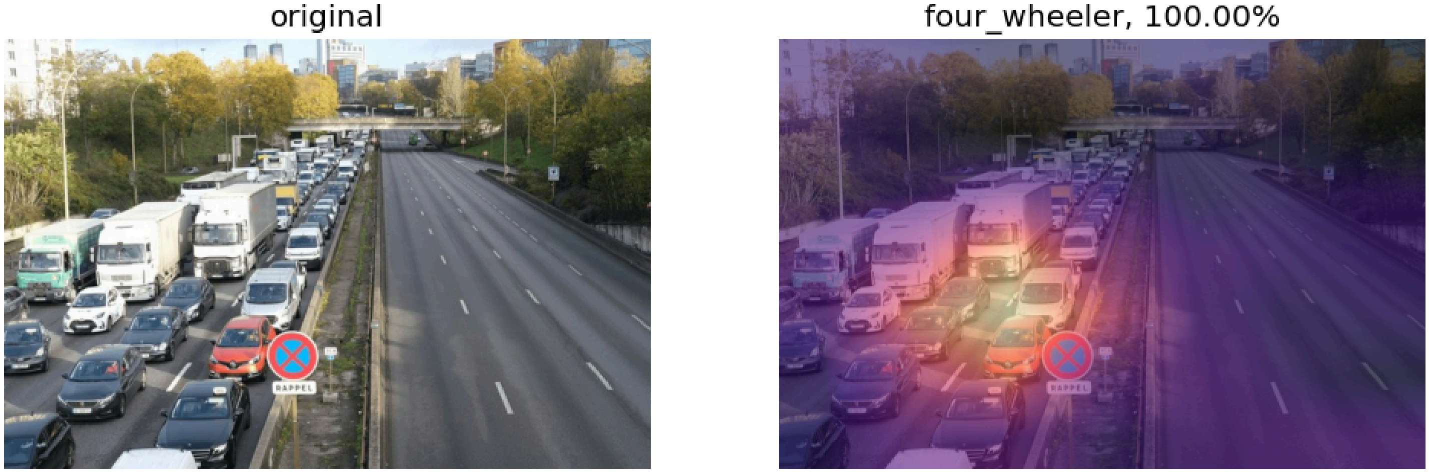
FastAI itself does not make the inference or training fast but the development is by far the fastest if we consider other frameworks while training a new model. We are basically interested in making a prototype rathar than a full-fledge solution. So, for this rapid prototyping and developement scheme of our operation, we better use the FastAI and again I am bound to deliver the project with the due time. Thats why it is best for the business.

Aftermath of Using the library

It made the development way faster. When development speed is fast, the debugging and feature implementation also tends to get influenced and be faster alongside. So, I felt that this way of training the model for just implementing a simple idea is the best.

What else is good about the trainign session?

The good thing is that I ran a GradCam session and found out the focusing area of the learner without any special preparation.



original

four_wheeler, 100.00%

So, here above 7you can see that the original image is of all four wheelers where the gradcam is focusing is also on the section of the four wheelers. Although it is not that accurate but isnt this the point? We want to see the discrepancy and improve.

YOLO model

YOLO is an updated version of SSD. Basically in yolo you can easily detect the object. And in the code we have basically used the module by `ultralytics` s YOLO implementation which is SOTA of its time. The `training-1-yolo.ipynb` file is reponsible for training the full model with 100 epoch.

And again, the `yolo_inference.py` file gives you the commands in commented format.