Overview:

We introduce a solution that's built upon using YoloV7 object detection algorithm to detect the visual pollution scenarios. Our object detection algorithm detects the visual pollution through the provided dataset where we passed through some stages like data exploration , data cleaning, data preprocessing , augmentation , training , validation and testing stages to achieve mAP of 77.43% , recall 77.04% and 68.22% precision. Our solution allowed fast, accurate real-time detection of the visual pollution signs in the streets.

Challenges

We faced too many challenges regarding solving the challenge Theme-1 , including data imbalancing as shown in figure 1, outliers , scaling bounding boxes , training time and mAP . To solve these challenges we have passed through some processes

1-removing outliers : we have removed the class BAD_streetLIGHT since it has only one training example which leads to data imbalancing problem and therefore noise

2-during Exploratory data analysis : we scaled the bounding boxes by 2 so it can fit the problem

3-Data augmentation : we used data augmentation to have various scenarios to the detection including rotation by 45 degrees , 25 brightness

4-During preprocessing : during preprocessing we stretched the image to have the dimensions of 512x512 and applied to the bounding boxes since yolov7 uses resnet50 as the backbone classifier

5-Training : the dataset was splitted into train, validation and test with ratios of 90%-8%-2% resulting in 8.5 k images for training , 751 images for validation and 190 images for test which resulted into mAP 77.43% , recall 77.04% and precision 68.22% as shown in figure 2

6- training time : we trained the model after several experiments on 740 epochs which took 35 hours on Nvidia RTX 2080s

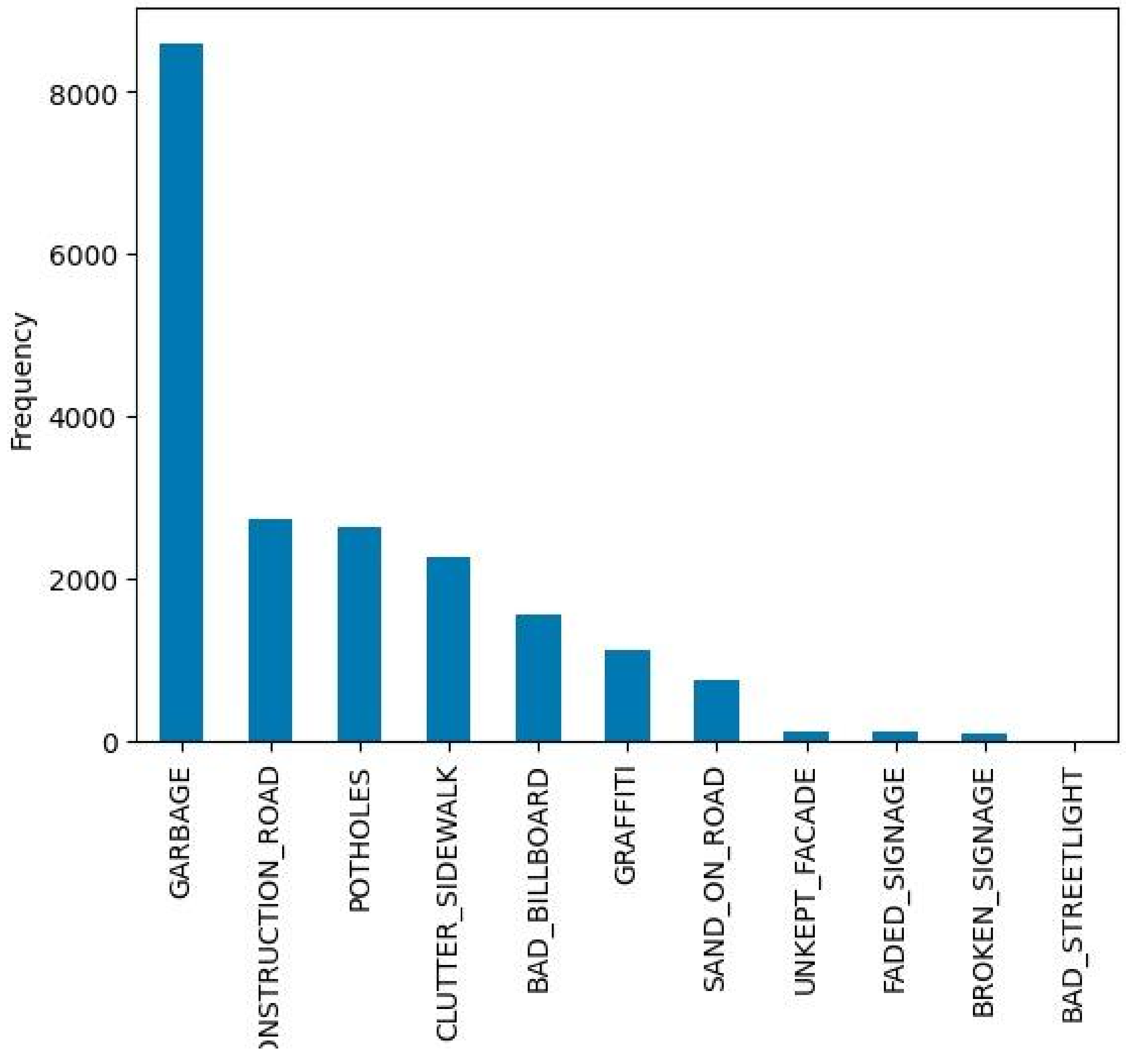

Figure 1 that shows the data imbalance problem

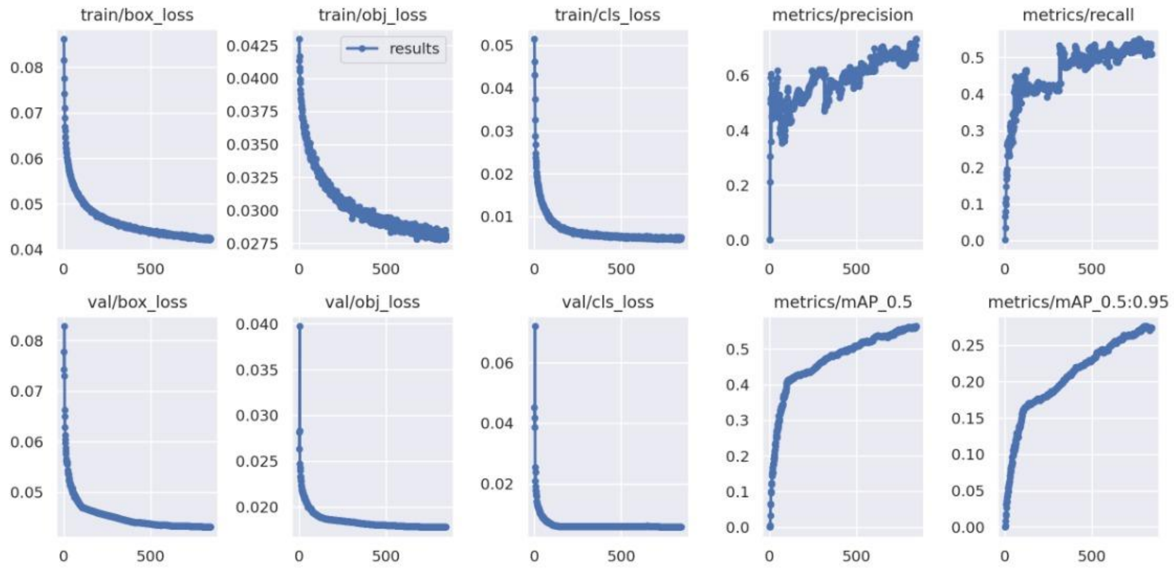

Figure 2 that shows the training outcomes to achieve the mAP 77.43%

Steps to make the solution more scalable

1-Collect and annotate more data regarding some classes like unkept_facade, broken_signage to avoid the biasing problem
2-provide computing infrastructure to allow continuous real time automated training
3-provide human annotators for more continuous data annotations
4-provide monitoring unit that's responsible for real time monitoring and logging
5-fixed real time streaming cameras either on the vehicles or on the streetlights to monitor visual pollutants
6- centralized hub to collect and process the real-time streaming on the spot then reply back with the prediction results so it can give an alarm about unusual events

Open Sources software :

1-pyTorch : for building and training the model
2-scikit-learn : for data splitting
3-numpy and pandas : for reading the CSV files and preprocessing the annotation and classes
4-OpenCV : for reading the images and preprocessing the images
5-Tensorboard : for visualizing the results of the training

To do if we have more time / data / resources

1- try more data annotations
2- dataset inject to increase and address the problem of data imbalance
3-Upgrade to more powerful computational power if cloud providers like aws and GCP are not favorites and we used to train locally
4- study the atmosphere of KSA widely and the frequency of visual pollution events occurrence
5- try more backbone architectures instead of ResNet50