Vehicles Detection

1st Hedaya Elattar 3092735
Stirling, UK
hae00048@students.stir.ac.uk

I. Introduction

Object detection is a common computer vision task, and it is used in tackling many problems such as traffic analysis and monitoring, and people counting [1]. In this project, the main focus is on vehicles detection including cars, motorcycles, buses, and trucks for wild images at street-level. The dataset consists of images from two cities which are Cairo and Stirling. Consequently, many experiments were held to cover different scenarios of model training on this dataset. In the following sections, more details about the proposed solution and results will be provided.

II. Proposed Solution with Justifications

A. Data creation

The dataset was collected using the sequences ids of Cairo and Stirling frames from Mapillary API [2]. Images were annotated, using Computer Vision Annotation Tool "CVAT", to include bounding boxes "xmin, ymin, xmax, ymax", object class, and filter label for the repeated, low quality, or empty images to be removed from the final dataset [3]. Consequently, 202 images from Cairo and 200 images from Stirling, and their PASCAL VOC format annotations were included in the final dataset. In terms of categories appearance, Cairo dataset includes 809 cars, 49 motorcycles, 15 buses, and 57 trucks, while Stirling dataset contains 548 cars, 2 motorcycles, 7 buses, and 19 trucks.

B. Proposed solution

In the pre-processing phase, images were read using OpenCV in an RGB format and resized. Due to the small size of the dataset, data augmentation was implemented such as horizontal flip, brightness and contrast change, fog addition, and Gaussian blur to increase the number of images and avoid over-fitting.

The dataset was divided into training, validation, and testing. Then, Faster Region-based Convolutional Network "Faster R-CNN", which is an anchor-based algorithm, was trained on the training set and validated on the validation set. Faster R-CNN was utilized with ResNet50 as a pre-trained backbone because it proved better mean average precision "mAP" and fatser inference time than VGG-16, and because ResNet101 is computationally expensive compared to ResNet50 [4]. Although You Only Look Once "YOLO" model is faster than Faster R-CNN, some researches shows, YOLO has some limitation such as that, it uses only two bounding boxes with one class per the grid cell, which can lead to spatial constraints [5]. This can affect the prediction ability of the model for nearby objects [5]. Moreover, it struggles with the prediction of objects in unfamiliar aspect ratios [5]. Therefore, the models tuning was conducted using Faster R-CNN and one of TorchVision library [6]. Regarding the hyper-parameters tuning, Grid search was used to find the highest performance of all permutations of learning rate, momentum, and weight decay values. Finally, Non Maximum Suppression "NMS" was employed in the filtering of the best predicted bounding box in terms of the Intersection over Union "IoU" threshold.

III. Results

There were three experiments of testing the ability of model to generalize and to predict the bounding box and its assigned class label. The first one in which the model was trained on 50% of Cairo and 50% of Stirling images and tested on the rest of them. In the second experiment, images of each city was split into training and testing, so that, the model trained on Cairo can be tested on city Cairo, and the model trained on Stirling can be tested on Stirling. The final experiment is dividing the data according to the city. Therefore, the model trained on Cairo can be tested on Stirling, and vice versa. "Fig. 1" shows the loss values during training on Cairo in experiment 3. The trained model was validated and tested by

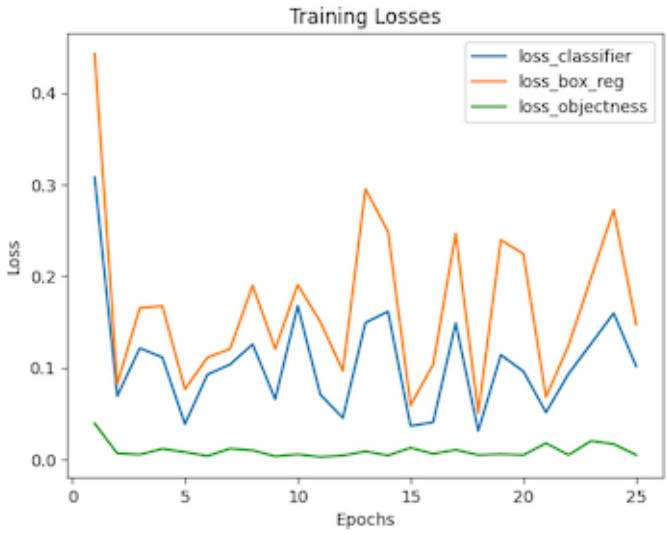

Fig. 1. Loss graph during third experiment training

mAP within different thresholds since it is one of the most utilized metrics in object detection problems. Accordingly, it is proved that, Faster R-CNN provides promising results in

Fig. 2. Sample from expected output and model output

vehicles detection, as shown in "Fig. 2", and it can generalize because the validation mAP is close to the testing mAP. For example, mAP value, which represents mAP for boxes with threshold from 50 to 95, in the first experiment is 0.2039, in the second experiment, when trained on Cairo, is 0.1511 and 0.2341 on Stirling. In the last experiment, it was 0.1924 and 0.1709 when trained on Cairo and Stirling respectively, as shown in "Table. I". However, because of the imbalanced classes, the model does not perform well on low appearance classes such as buses and motorcycles.

TABLE I
EXPERIMENTS RESULTS

Experiment Name	map	map50	map_75	map per class			
Experiment_1	0.2039	0.3394	0.1818	0.5624	0.067	0.0744	0.1117
Experiment_2_Cairo	0.1511	0.2491	0.1595	0.5336	0.0355	0.0149	0.0204
Experiment_2_Stirling	0.2341	0.3929	0.2315	0.5961	0	0	0.3402
Experiment_3_Train_Cairo	0.1924	0.2938	0.209	0.5821	0.0017	0.025	0.1609
Experiment_3_Train_Stirling	0.1709	0.2705	0.1979	0.5243	0.0062	0.1255	0.0278

IV. DISCUSSION

From the aforementioned research and development, Faster R-CNN generates good results, from 0.1511 to 0.2341 mAP, in vehicles detection and can generalize in different cities, which means this architecture can be used in real-time applications like autonomous driving, and traffic analysis. Despite this, it can be improved by collecting more data especially for rare classes, or by using other architecture like YOLO after data re-balancing with class weight or focal loss parameters [7].

V. CONCLUSION

In conclusion, vehicles detection has important real-life applications. That is why, Faster R-CNN tuned on a collected dataset from Cairo and Stirling. This dataset includes four classes of vehicles. Three main experiments were conducted to analyze the localization and classification ability of the model. These trials were measured by mAP, which proves promising results from Faster R-CNN in this problem, but still the model needs more investigation by collecting more data to fix the problem of imbalanced classes in the dataset such as motorcycles and buses.

REFERENCES

[1] J. Deng, X. Xuan, W. Wang, Z. Li, H. Yao, and Z. Wang, "A review of research on object detection based on deep learning," in Journal of Physics: Conference Series, IOP Publishing Ltd, Nov. 2020. doi: 10.1088/1742-6596/1684/1/012028.
[2] Mapillary. [Online]. Available: https://www.mapillary.com
[3] Computer Vision Annotation Tool (CVAT). (2.2.0), CVAT.ai Corporation. [Online]. Available: http://cvat.ai/
[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.01497.
[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.02640.
[6] TorchVision: PyTorch's Computer Vision library. (2016-11-06). TorchVision maintainers and contributors. [Online]. Available: https://github.com/pytorch/vision
[7] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance Problems in Object Detection: A Review," Aug. 2019, [Online]. Available: http://arxiv.org/abs/1909.00169.