Project 1: Object detection techniques (in case of small objects) on AU Drone dataset.

Kushalkumar Suthar
Btech in CSE
Ahmedabad University
kushalkumar.s@ahduni.edu.in

Dhruvin Prajapati
Btech in CSE
Ahmedabad University
dhruvin.p@ahduni.edu.in

Rohit Rathi
Btech in CSE
Ahmedabad University
rohit.r1@ahduni.edu.in

Nish Parikh
Btech in CSE
Ahmedabad University
nish.p@ahduni.edu.in

***Abstract*—Detection of objects in computer vision is crucial for a variety of practical uses, such as autonomous driving and surveillance systems. Even with notable progress in deep learning, spotting tiny objects is still difficult because of downsampling, differences in receptive fields, and sensitivity to changes in bounding boxes. By conducting thorough experiments and assessments on standard datasets such as Visdrone-2019, we discover unexpected observations regarding the efficiency and effectiveness of the Faster-RCNN and RetinaNet models. Our results definitively show that Faster R-CNN outperforms RetinaNet in terms of detection accuracy, despite what was expected. This discovery emphasizes the long-lasting effectiveness of Faster R-CNN in the field of object detection. Our research adds important perspectives to the current conversation on object detection methods and sets the groundwork for future studies in this area.**

***Index Terms*—Faster-RCNN, RetinaNet, mAP, Small Object Detection, High-Resolution Features, Feature-Pyramid Based Object Detectors, Inference Speed, VisDrone Dataset.**

I. Introduction

IN the current day, there are numerous methods to detect objects in the computer vision field, from the various deep learning models like QueryDet, MRDet, ReDet, HoughNet, EfficientDet, HRDNet, MatrixNets, SyNet, RetinaNet, and CNN models, to handcrafted feature-based methods like HOG and SIFT. Choosing one from these is a difficult decision to make without knowing the accuracy and efficiency of the dataset for each of the models. To find the more accurate model we have compared the RetinaNet model with Faster-RCNN on the Visdrone-2019 dataset. In both of the models, we have used ResNet as a backbone architecture and compared the performance metrics which are denoted by mAP (Mean Average Precision) scores.

II. Methodology

Faster R-CNN, an extension of the R-CNN and Fast R-CNN fashions, sticks out for its robustness in dealing with small items inside cluttered backgrounds. Its architecture revolves round a -diploma gadget, integrating the Region Proposal Network (RPN) and the subsequent Fast R-CNN detector. This setup gives a entire framework that successfully addresses the worrying conditions posed thru small item detection. The RPN plays a pivotal position by the use of successfully producing place proposals, permitting the version to concentrate its interest on potential item places. This feature is mainly incredible in conditions in which small gadgets might in all likelihood resultseasily move ignored amidst complex backgrounds. Furthermore, Faster R-CNN's capability to refine and classify those proposals contributes extensively to improving localization accuracy, a crucial problem of precisely detecting small gadgets. However, no matter its efficacy, Faster R-CNN's complexity and relatively slower inference velocity pose ability limitations, especially in applications necessitating real-time processing of small items.

Contrarily, RetinaNet emerges as a compelling answer tailor-made for small item detection obligations, way to its simplicity and effectiveness. The version's structure follows a unmarried-diploma approach, complemented by means of using the usage of the Feature Pyramid Network (FPN). This setup allows the green detection of items during more than one scales, together with small ones Unlike Faster R-CNN, RetinaNet bypasses the want for a separate perception era step, at once predicting bounding boxes and sophistication opportunities. This streamlined machine, coupled with the present day focal loss characteristic adept at managing beauty imbalance, yields aggressive accuracy even for small devices. Moreover, RetinaNet's faster inference pace in addition enhances its appeal, in particular in applications in which fast detection of small gadgets is paramount, collectively with surveillance or clinical imaging.

III. Dataset and Evaluation Metric

In this work, we have used the Visdrone-2019 dataset. The Visdrone-2019 dataset consists of 6,471 images for training, 548 images for validation, and 3,190 images for testing. The images that the dataset contains are taken by drone-mounted cameras from 14 cities in China. Moreover, the dataset gives annotation files consisting of x and y coordinates of the boundary box, height and width of the boundary box, score, truncation, and the category ID for each of the objects present in each image. Additionally, the objects in the dataset are classified into 10 classes.

Image 1

Image 1 is the example image from the Visdrone dataset.

ID	xc	yc	w	h	Score	Category ID	Trun.
3	777	33	53	1	10	0	0
96	783	33	37	1	2	0	0
150	827	46	57	1	1	0	1
186	833	37	60	1	1	0	2
263	697	18	56	1	1	0	1

Table 1: Given dataset

Table 1 contains 5 object details of the image 1. In Table 1, the ID column contains object ID, xc and yc are x and y coordinates of the boundary box around that object, and w and h are the width and the height of that boundary box, and trun. refers to truncation.

To evaluate both models we calculated their AP, AP50, AP75, and AP95 scores, where AP is the average of ten IoU thresholds from 0.5 to 0.95, and AP50 is the average value with the IoU threshold of 0.5. Similarly, AP75 is the average value with the IoU threshold of 0.75, and AP95 is the average value with the IoU threshold of 0.95.

A. *RetinaNet*

Image 2: RetinaNet

Model	Backbone	AP[%]	AP50[%]	AP75[%]	AP95[%]
RetinaNet	ResNet-50	35.85	41.19	36.26	27.14

B. *FasterRCNN*

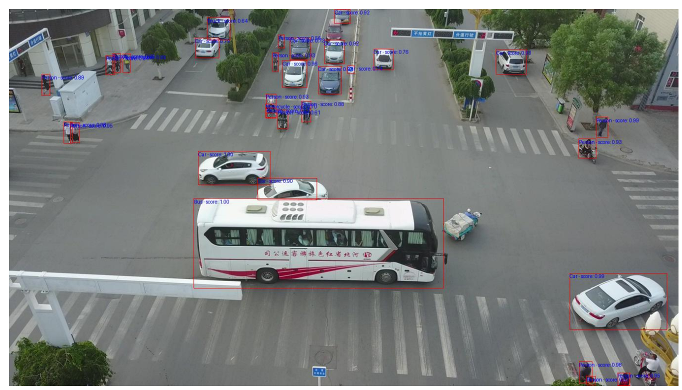

Image 3: FasterRCNN

Model	Backbone	AP[%]	AP50[%]	AP75[%]	AP95[%]
Faster -RCNN	ResNet-50	49.53	62.13	48.68	33.98

IV. Results

We have implemented two models which are Faster-RCNN and RetinaNet on the val set of the visdrone dataset. We have implemented it on 100 out of 584 files and found out that Faster RCNN has better performance than RetinaNet in terms of mAP/AP[%] values as shown in the table.

V. Discussion

We have implemented the code on 100 files out of 584 due to two reasons, first is we want to validate the model in terms of selected 100 files (unseen data) to compare the performance and second is that we have resource constraints of the device on which the model was running. So based on the above discussion of the dataset used, we want to then compare the performance of Faster RCNN and RetinaNet on the dataset.

VI. Summary

So, there are many models for Object Detection, but all are in process to achieve a better accuracy for small object detection. Through our project, we wanted to implement and find out and compare the performance of two models, namely Faster RCNN with ResNet50 and RetinaNet. And through our work, we found out that Faster RCNN to be a better model than RetinaNet.

VII. References

Author links open overlay panelOnur Can Koyun a et al., “Focus-and-detect: A small object detection framework for aerial images,” Signal Processing: Image Communication, https://www.sciencedirect.com/science/article/pii/S0923596522000273 (accessed Apr. 8, 2024).

https://developers.arcgis.com/python/guide/faster-rcnn-object-detector/
https://developers.arcgis.com/python/guide/how-retinanet-works/: :text=object