Object Classification in Urban Aerial Drone Footage

Jack Chin, Liam McGoldrick, Aidan Walker

***Abstract* — This paper details a deep-learning computer vision architecture for object classification of aerial drone footage in urban environments. We developed and evaluated two models on the 2019 VisDrone dataset, which comprises over 260,000 frames of footage from various cities in China. Our baseline model utilized a pretrained Faster R-CNN architecture with a ResNet-50 backbone, modified to detect VisDrone-specific object classes. Our custom model introduced the following optimizations to this baseline: custom anchor sizes for the bounding boxes within the Region Proposal Network (RPN), advanced Region of Interest (RoI) pooling for multi-scale feature maps, and dropout regularization of a new MLP head. We evaluated both models using mean Average Precision (mAP), precision, recall, and F1. The custom model achieved [metric] improvements over the baseline. This work highlights the strength of robust pre-trained models, and the limitations of certain optimization techniques, particularly with GPU constraints.**

***Keywords* — Deep Learning, Computer Vision, R-CNN, Object Detection, VisDrone**

I. Introduction

Drones and other unmanned aerial vehicles (UAVs) are becoming increasingly utilized in all areas of life: surveillance, emergency response, environmental monitoring, etc. Many of these use cases require the effective processing of large-scale aerial footage for automated scene understanding. However, in urban environments with high object density and variation, many traditional computer vision techniques struggle. Deep learning provides a promising solution, depending on dataset quality, model architecture, and resource constraints.

This paper focuses on object classification, specifically to identify and localize multiple object types such as cars, pedestrians, and bicycles using real-world drone footage. While previous work has focused on very specific settings (such as wildlife or a particular urban area), our project leverages a much more varied and diverse dataset in VisDrone. The rest of this report covers previous work, dataset characteristics, model design, evaluation metrics, experimental results, and conclusions.

II. Previous work

As interest and investment in drone application has increased, so too has the academic research into the field. We primarily referenced two main academic sources in developing our project: Stanford Drone Dataset (SSD) and a SeaDronesSee Dataset.

A. Stanford Drone Dataset

Introduced by Robicquet et al. in 2016, the SDD is a comprehensive dataset of drone footage of pedestrians, bicyclists, skateboarders, cars, buses, and golf carts navigating a Stanford University. The dataset (and the research utilizing the dataset) has been primarily leveraged for studying social behaviors in crowded environments and pedestrian movements, but still contributes greatly to understanding our specific object detection task, especially given the urban pedestrian focus. However, it is limited in that a fairly small number of classes are represented in the overall dataset (six).

B. SeaDronesSee Dataset

This project aimed to fine-tune a Faster R-CNN model for the SeaDronesSee dataset, which comprises ocean aerial images. This dataset was compiled for use in Search and Rescue contexts – thus the small-scale object detection needed for the task was perfectly suited for a Faster R-CNN application. The researchers also utilized various optimization techniques including image patching to more efficiently complete training on large, high definition training data.

III. Dataset

We utilized the VisDrone 2019 dataset—a large-scale competition dataset specifically designed for object detection in drone footage. The dataset was developed by researchers from Tianjin University, and comprises over 260,000 video frames and 10,000 static images taken in urban and rural environments across 14 cities in China. 11 distinct object categories are represented within the dataset, with a disproportionate percentage of objects being pedestrian and van objects, per Figure 1.

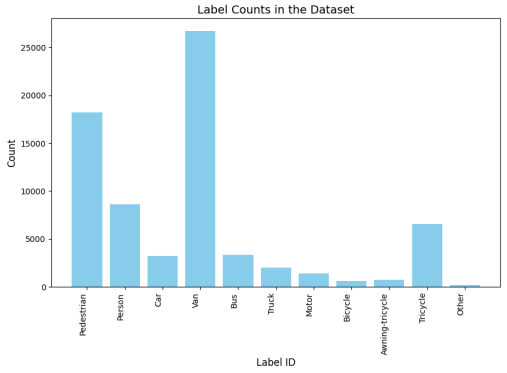

Fig. 1 A histogram showing the distribution of object labels within the VisDrone 2019 dataset

The dataset is very well maintained, so minimal pre-processing techniques were necessary. We first removed "ignored region" labels to avoid training bias. We then standardized image sizes, making sure to scale the bounding box locations accordingly. We also analyzed the distribution of image complexity, and removed outliers with significantly higher and lower number of objects. We then created a custom data loader to execute data augmentation and select a subset of the overall dataset (for more efficient training).

Fig. 2 Visualized sample image from the VisDrone dataset showing bounding boxes and labels

IV. Method

We developed two deep learning models for this project – a custom model utilizing advanced optimization techniques, and a baseline model to compare evaluation metrics against.

A. Baseline Model

Our baseline model utilized a pre-trained Faster R-CNN architecture with a ResNet-50 backbone, trained on the COCO image-detection dataset (330,000+ diverse image classes and contexts). The model utilizes a Feature Pyramid Network (FPN) and an advanced Region Proposal Network (RPN), with over 40 million trainable parameters. We then replaced the classifier head to focus solely on the 11 relevant labels from the VisDrone dataset.

B. Custom Model V1

Our Iteration V1 model builds upon the baseline by upgrading the backbone from ResNet-50 to ResNet-101, which means it now has more layers to learn from and can capture more detailed features in images. The Region Proposal Network (RPN) and training hyperparameters were kept consistent to isolate the performance impact of the deeper backbone.

C. Custom Model V2

In Iteration V2, we returned to the ResNet-50 backbone but introduced several training optimizations to improve performance. We fine-tuned the learning rate and added a learning rate scheduler to help the model converge more smoothly during training. Additionally, we increased the number of region proposals to give the model more candidate object detections to improve accuracy.

D. Custom Model V3

In Iteration V3, we continued using the ResNet-50 backbone and added dropout layers to the region of interest (RoI) head to reduce overfitting and improve generalization. Additionally, we customized the anchor generator with smaller sizes to better capture distant objects and adjusted the RoI pooling to handle scale variation more effectively.

V. Evaluation

We modeled our evaluation metrics based on the VisDrone competition official criteria, which focused on four main measures: mean average precision (mAP), average precision, average recall, and f1. We used mAP and precision to quantify the accuracy of the bounding box size and locations by applying different Intersection over Union (IoU) thresholds starting from 0.50 to 0.95 in increments of 0.05. We used recall to quantify the model's ability to find all the relevant objects in a given image (the "completeness" of the detection). We

calculated these metrics both for the models' overall performance as well as for individual class labels.

VI. Results

Our initial baseline model achieved a mAP score at IoU threshold 0.5 of 0.1145, and a mAP score at IoU threshold of 0.75 of 0.0459. This sharp drop indicates the baseline model's difficulty in precise localization, as it is significantly better at classifying when more loose bounding boxes are allowed.

Fig. 3 Visualized bounding box and label predictions from baseline model, confidence threshold of 0.3

As we can see in figure 3, the quantity of predictions is quite high and many of the objects are classified correctly despite confidence scores frequently below 0.5. Ultimately, the baseline model predicts with low confidence scores but detects objects decently well.

The first iteration model achieved a mAP score at IoU threshold 0.5 of 0.0095, and a mAP score at IoU threshold of 0.75 of 0.0010. These results show a 92% and 98% decrease respectively from the baseline model results for these metrics. This decrease in performance resulted in even lower confidence predictions, with none surpassing the threshold of 0.3 Despite the low accuracy of the model, we still don't often misclassify objects (according to a confusion matrix analysis done on our test set).

The second iteration of the model showed a significant improvement from the first iteration, but is still in many metrics a 3 times decrease in performance from the baseline model. It achieved a mAP score at IoU threshold 0.5 of 0.0308, and a mAP score at IoU threshold of 0.75 of 0.0018. This improvement from iteration one can be clearly seen in figure 4, in which we can observe many predictions with above 0.3 confidence score. However, we can note the significant decrease in quantity of predictions, as many objects are missed or had very low confidence predictions. We saw roughly a 50% decrease in F1 scores for all classes between the baseline and iteration two, from 0.4243 to 0.2857 for pedestrians, and 0.5651 to 0.2714 for vans. However, we once again observed relatively few misclassifications in our confusion matrix analysis.

Fig 4. Visualization of predictions made by the second iteration V2 model.

In the third iteration of the model, we observed another significant decrease in performance from the baseline model, and even the second iteration. The third iteration achieved a mAP score at IoU threshold 0.5 of 0.0170, and a mAP score at IoU threshold of 0.75 of 0.0013, which represent about a 50% decrease in performance from the second

iteration. We observed a significant decrease in recall and precision for classes other than pedestrians. This change is reflected in figure 5, which shows how the third iteration model was only able to predict pedestrians with a confidence score of above 0.3, and had very poor performance when predicting other classes of objects.

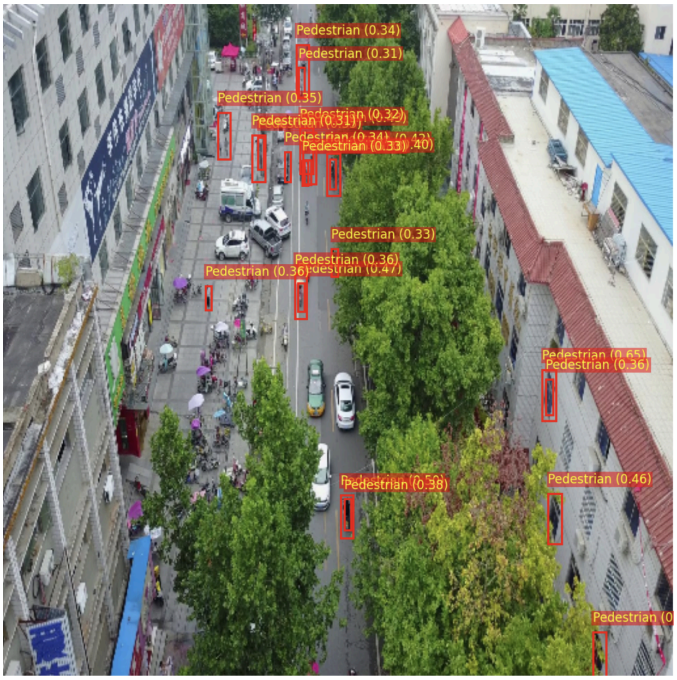

Fig 5. Visualization of predictions made by the third iteration V3 model.

VII. Discussion

Overall, we had nonlinear progress with our three training iterations of the model. We achieved our best results with the baseline model, but similar results with the second iteration of the model. This was most likely due to using the ResNet50 backbone, which ended up being much more accurate than the ResNet101 backbone that we used in the first iteration of the model. While in theory better suited for our application, the ResNet101 backbone was likely too complicated and was unable to train effectively on the small amount of training data we were required to use due to GPU constraints. The ResNet50's simplicity was likely suited much better for the circumstances of our training data than the ResNet101 backbone. The decreased performance in the third iteration, which used the ResNet50 backbone, could be explained by the addition of dropout layers. In conjunction with the limited number of training images, the dropout layers could have been counterproductive and made the already stunted training process even less effective. Overall, our results potentially show the effectiveness of the ResNet50 backbone on our application, but due to the GPU constraints, it remains unclear if the ResNet101 backbone could have been more effective given more time to train. Similarly, we observed a decrease in performance with the addition of dropout layers, but again it is unclear whether these would have contributed to an increase in performance given more training.

VIII. Summary

This paper explored the application of deep learning model architectures to object classification in complex, dense, urban drone footage. We developed and evaluated two models: a baseline pre-trained Faster R-CNN model and a custom model with optimizations tailored for the unique challenges of our chosen dataset, VisDrone 2019. Our results showed that the targeted improvements to the custom model did not yield significant improvements in performance. While our project faced considerable obstacles concerning computational constraints and overall dataset complexity, we still gained meaningful insight into the effectiveness of iterative, targeted optimizations on already robust pre-trained models and the importance of utilizing well-maintained, varied, accurately labeled training data. There are many future directions for this research, including training on broader datasets for greater generalization, applying temporal modeling across video sequences, and integrating additional input data such as infrared or elevation metadata.

References

[1] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and Tracking Meet Drones Challenge," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 7380–7399, Nov. 2022, doi: 10.1109/TPAMI.2021.3119563.

[2] A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, *Learning Social Etiquette: Human Trajectory Prediction In Crowded Scenes* in European Conference on Computer Vision (ECCV), 2016.

[3] Mou, C.; Liu, T.; Zhu, C.; Cui, X. *WAID: A Large-Scale Dataset for Wildlife Detection with Drones*. Appl. Sci. 2023, 13, 10397. https://doi.org/10.3390/app131810397

[4] J. Jaykumaran, "Fine-tuning Faster R-CNN on Sea Rescue Dataset – Small Object Detection: PyTorch," *LearnOpenCV*, May 28, 2024