

Indian Road Traffic Analysis: Using UAV Imagery and Deep Learning Techniques on Custom Annotated Dataset

A Project Report

Submitted by:

Anshi Shah (AU2040087)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE

at



**Ahmedabad
University**

School of Engineering and Applied Sciences (SEAS)

Ahmedabad, Gujarat

May, 2024

DECLARATION

I hereby declare that the project entitled '**Indian Road Traffic Analysis: Using UAV Imagery and Deep Learning Techniques on Custom Annotated Dataset**' submitted for the B. Tech. (**Computer Science and Engineering**) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Anshi Shah

Date:

Place:

CERTIFICATE

This is to certify that the project titled '**Indian Road Traffic Analysis: Using UAV Imagery and Deep Learning Techniques on Custom Annotated Dataset**' is the bona fide work carried out by **Anshi Shah**, a student of B. Tech. (**Computer Science and Engineering**) of School of Engineering and Applied Science at Ahmedabad University during the academic year 2023-2024, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science and Engineering** and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Prof. Mehul Raval

Date:

Place:

Acknowledgement

I want to thank everyone who has helped me with my thesis project during my undergraduation. Professor Mehul Raval, my thesis faculty advisor has always been there to support and guide me throughout thesis work. His knowledge, insights and patience have really helped improve the quality and development of this thesis work for research. I am glad to have his guidance and insightful solutions for completing this work. I would like to thank Mr. Yagnik Bhavsar for his expertise and unwavering assistance whenever I required it. His advice, throughout the project phases has underscored the value of his mentorship and the significant dedication he has put into supporting me.

I want to express my appreciation to the School of Engineering and Applied Sciences, at Ahmedabad University for providing the support and resources for our research project. Institutional backing and resources play a role in the execution of this project. I also extend my gratitude to the developers and contributors of the open source tools, libraries and applications utilised in this project. Their dedication to creating and maintaining these resources greatly facilitated the progress and analysis of our study. I am also thankful to my family and friends for their support, understanding and motivation in all my endeavours. Their constant belief in me drives me forward. A big thank you to everyone involved in this thesis project. Your guidance and support have been invaluable to me.

Abstract

Traffic congestion experienced in almost all the major cities is mainly caused by urbanisation. The increasing number of vehicles on the roads is the cause. Urban planning plays a role in developing transportation systems, which heavily relies on assessing traffic patterns. Managing traffic regulations can be quite intricate in India due to the paced and constantly changing flow of vehicles beyond monitoring systems. This research project aims to analyse road traffic dynamics using drone footage and advanced deep learning algorithms with a custom dataset of Ahmedabad.

The primary goal of this study is to establish a groundwork for recognizing and analysing objects in images captured by drones. Using a DJI Mavic 2 Pro drone resolution aerial videos of a roundabout were obtained, serving as the specific dataset for my research. Detailed annotations were meticulously added at the level including bounding boxes, box orientation and labels for entities like vehicles and pedestrians within the dataset. Cutting edge deep learning models such as YOLOv8 and BBA Vectors (Box Boundary Aware Vectors) were also employed to achieve outcomes in object recognition tasks based on the custom dataset. The results highlight its performance and its capability to detect and categorise objects, in conditions.

The outcome of this project will have an impact on planning and traffic control in India. The current efforts, toward enhancing this system are focused on enhancing traffic flow enhancing road safety and helping the decision making by understanding traffic patterns, vehicle distribution and enforcement of traffic regulations.

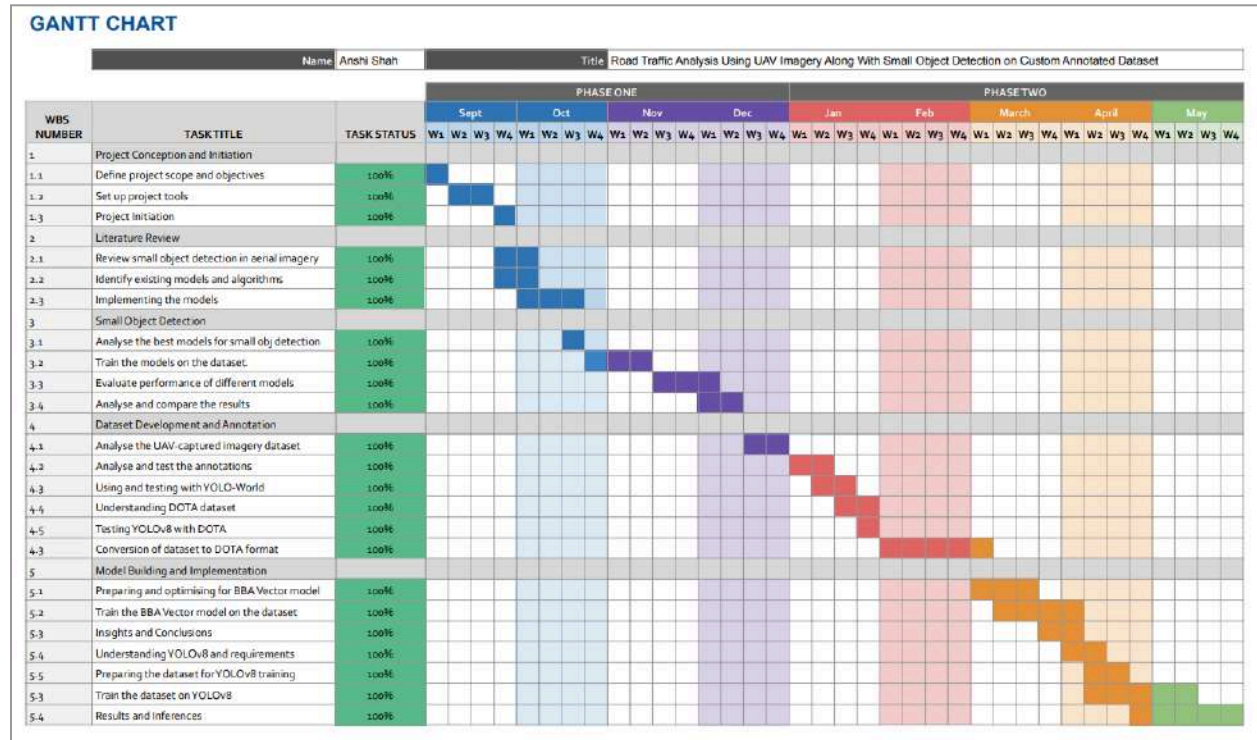
Table of Contents

Declaration.....	2
Certificate.....	3
Acknowledgement.....	4
Abstract.....	5
Table of Contents.....	6
List of Figures.....	7
Gantt Chart.....	8
Introduction.....	9
1.1 Project Definition.....	10
1.2 Project Objectives.....	10
Literature Review.....	12
2.1 Related Work.....	12
2.2 Tools and Technologies.....	13
Methodology.....	14
3.1 Normalised Wasserstein Distance (NWD) Implementation.....	14
3.1.1 VisDrone Dataset.....	14
3.1.2 Model Building.....	14
3.1.3 Model Results.....	15
3.2 AU Drone Dataset Preparation.....	17
3.2.1 AU Drone Dataset.....	17
3.2.2 Dataset Annotations.....	18
3.2.3 Using and Testing with YOLO-World.....	19
3.2.3 DOTA Dataset.....	20
3.2.4 Testing YOLOv8 with DOTA.....	21
3.2.4 Conversion of Annotations.....	24
3.3 Model Building.....	26
3.3.1 Box Boundary-Aware Vectors.....	26
3.3.2 YOLOv8.....	30
Results.....	34
4.1 BBA Vectors.....	34
4.2 YOLOv8.....	36
Conclusion.....	42
5.1 Project Outcomes.....	42
5.2 Real-World Applications.....	43
5.3 Future Work.....	44
Bibliography.....	46

List of Figures

Figure	GANTT Chart
Figure 3.1	IoU-Deviation Curve
Figure 3.2	NWD-Deviation Curve
Figure 3.3	YOLO-World Results
Figure 3.4	DOTA Annotation Format
Figure 3.5	DOTA Annotations Drawn to Image
Figure 3.6	YOLOv8 Interference on DOTA Image
Figure 3.7	Verifying sConverted DOTA Annotation
Figure 3.8	Verifying Converted DOTA Annotations
Figure 3.9	Architecture of BBA Vector Model
Figure 3.10	BBA Vector Method
Figure 3.11	Architecture of YOLOv8
Figure 4.1	BBA Vector Result (Frame Comparison)
Figure 4.2	BBA Vector Results (3-Wheelers)
Figure 4.3	Validation Batch Labels
Figure 4.4	Validation Batch Prediction
Figure 4.5	YOLOv8 Results (Metrics)
Figure 4.6	YOLOv8 Results (mAP)
Figure 4.7	YOLOv8 Results (Confusion Matrix)
Figure 4.8	YOLOv8 Results (Class Based)

GANTT CHART



Chapter 1

Introduction

India has one of the most complex and congested traffic conditions. The road system in India reflects its population. The traffic conditions in India are constantly changing as cars, trucks, motorcycles and auto rickshaws all use the roads[1]. The growth of cities and improvements in the economy have led to vehicles on the roads creating a complex situation[1].

Analysing traffic in India holds immense importance where the major cities in India are consistently among the most congested places all around the world[1]. This congestion leads to air pollution, increased fuel consumption which is resulting in economic losses[2].

New advancements in traffic studies have been facilitated by the emergence of Unmanned Aerial Vehicles (UAVs) and progress in Deep Learning methods. Detecting objects in images is a key technique that is being explored. Given the sizes and scattered distribution of objects in aerial photos, object detection poses a challenging yet crucial task[3]. Nonetheless, with the aid of learning techniques it is now achievable to accurately recognize and classify items depicted in these images[4][5].

The aim of this research is to analyse road traffic by utilising UAV images and advanced learning methods. The main emphasis is on creating a system for identifying objects in pictures and establishing a specialised labelled dataset.

The primary goal is to provide insights to aid in planning and traffic control ultimately contributing to the advancement of intelligent and eco-friendly cities in India.

1.1 Project Definition

The main aim of the project is to evaluate traffic situations on roads by utilising images captured by vehicles (UAVs). The AU Drone Dataset acts as the basis for this study. The goal is to establish a dataset, for studying vehicle behaviours and traffic flow. For the purpose of item detection in aerial photos, a ground truth must be created for the dataset, particularly in the setting of intricate and congested traffic scenes in India. Through the use of cutting-edge deep learning techniques, the initiatives ought to yield insightful data that will help India's urban planning and traffic management become more effective.

1.2 Project Objectives

Data collection and annotation:

High-quality aerial videos of Indian road traffic are collected using a DJI Mavic 2 Pro drone. The collected videos are preprocessed and frames are extracted. A custom dataset (AU Drone Dataset) is thus created, by labelling vehicles and pedestrians, using oriented bounding boxes and class labels.

Dataset preparation:

- Explore and understand the annotation format of AU Drone dataset. This consists of CSV files containing detailed attributes for each object of each frame.
- Convert the dataset annotations from its original format to DOTA format for increasing its compatibility with modern object detection models.

Model Implementation:

- Implement advanced deep learning models that are built for oriented object detection.
- Preprocess the dataset as per the requirements of the model for optimising the training efficiency.
- Train the selected models on AU Drone Dataset according to suitable hyperparameters.

Model Evaluation:

- Analyse the trained models and determine how well they can locate and identify items.
- Use appropriate metrics to analyse the data and assess how well the model handles different object classes and orientations.
- Determine the advantages and disadvantages of the models as well as our ground truth (the dataset), and suggest possible changes.

Applications:

- Draw valuable insights such as traffic patterns, vehicle distributions, traffic violations, etc.
- Discuss real-world applications like traffic management, urban planning for India.

Chapter 2

Literature Review

The use of deep learning techniques and UAV images has resulted in substantial breakthroughs in traffic analysis research. Deep learning techniques have improved the accuracy and efficiency of object detection in aerial photos. The use of drones (UAVs) has created opportunities for studying traffic patterns and urban planning. Past research has demonstrated the power of machine learning techniques, in identifying and classifying objects captured in photos offering perspectives for traffic control and city planning.

2.1 Related Work

Lately there has been a growing fascination with using the deep learning techniques and aerial footage from drones for studying traffic patterns. Several studies have explored the applications of these technologies, in scenarios.

BBAVectors-Oriented Object Detection [6]: This research incorporates a vector format designed for recognizing oriented objects in images, with precision. The approach combines representations and bounding boxes to accurately identify and classify oriented objects.

Oriented Object Detection Topics (GitHub)[7]: A selected collection of GitHub repositories and projects centred on object detection with a focus on object orientation.

Papers with Code - Oriented Object Detection [8]: There are a range of research papers and practical examples related to identifying oriented objects on this platform.

Papers with Code - Object Detection Models [9]: During the study, oriented object detection models and techniques are explored that could potentially be applied to identify oriented objects.

ArXiv Paper - "Single-Stage Oriented Object Detection with Learned Anchor Closeness" (2021) [10]: This study has an approach for identifying oriented objects by leveraging learned anchor proximity to enhance detection accuracy.

WACV Paper [6] - "Oriented Object Detection in Aerial Images With Box Boundary-Aware Vectors" (2021): This research paper discusses a method for identifying objects with orientations, in images focusing on handling issues related to varying orientations and aspect ratios.

YOLOv8.1 Model for Oriented Object Detection [11]: The latest iteration of the known YOLOv8 object detection model introduces elements and configurations tailored for identifying oriented objects in various applications.

2.2 Tools and Technologies

Deep learning frameworks: Tensorflow, PyTorch

Annotation tools / framework: YOLO-World

Programming languages and libraries: Python, OpenCV, NumPy, Matplotlib

Hardware: GPU (Quadro RTX 6000), CUDA (v. 11.4), NVIDIA-SMI (470.199.01)

Software and OS: Ubuntu, Google Colab, Jupyter Notebooks, Anaconda, Firefox

Chapter 3

Methodology

3.1 Normalised Wasserstein Distance (NWD) Implementation

3.1.1 *VisDrone Dataset*

The implementation is carried out using the VisDrone dataset. This dataset is a large-scale benchmark with annotated ground truth data for a range of computer vision applications. These are linked to image and video analysis using drones.

VisDrone consists of 10,209 images and 288 video clips totaling 261,908 frames that were taken using different drone-mounted cameras. Pedestrians, cars, bicycles, and other objects are among the many aspects included by the dataset, which also includes location (14 different cities around China), environment (urban and rural), and density (sparse and congested settings). The information was gathered in a variety of weather and illumination circumstances via a range of drone platforms. [12]

3.1.2 *Model Building*

Gaussian Modelling for Bounding Boxes: Since most real objects are not strictly rectangular, bounding boxes for small objects frequently contain background pixels. Foreground pixels are concentrated in the centre of these bounding boxes, whereas background pixels are concentrated on the perimeter. A 2-D Gaussian distribution is used to model the bounding box in order to more accurately represent the weights of the various pixels within it. The bounding box's centre pixel has the largest weight, and as one moves towards the edge, the pixel's significance diminishes.

Normalised Gaussian Wasserstein Distance (NWD): The Wasserstein distance is used to compare Gaussian distributions. Gaussian distributions can be compared using the Wasserstein distance. A normalised variant of this technique is the Normalised Wasserstein Distance (NWD), which was created to increase its usefulness. Scale invariance, location and deviation accounting, and similarity measurement even for non-overlapping bounding boxes are provided by NWD.

Model Building: The foundation is a pre-trained convolutional neural network (CNN), called ResNet in this case. From the supplied image, hierarchical characteristics are extracted. RPN uses the feature map that the backbone generates to function. From every region proposal, fixed-size feature maps are extracted by ROI pooling. It preserves spatial information by converting variable-sized ROIs into a fixed-size feature map. Regression and classification problems both make use of fully connected layers and extra heads. After classifying each ROI into object classes (such as human or car), the bounding box coordinates are refined.

NWD is used to assign the suggested bounding boxes to ground truth bounding boxes for training purposes during the training phase, which follows the receipt of region proposals from the RPN. Labels are assigned based on the NWD, which is computed for every proposed bounding box in relation to all ground truth boxes. Bounding boxes with high NWD values may be assigned a different label since they are thought to be less comparable to the ground truth. The bounding box assigner is an important tool that helps the model decide which of the suggested bounding boxes should be considered positive or negative samples. The model optimises its parameters based on both classification and regression losses.

3.1.3 Model Results

The limitations of the existing IoU functions are overcome by using the NWD based loss function. Despite the lack of overlap between the anticipated and ground-truth bounding boxes, gradients are however provided by NWD-based loss. This helps in strengthening the performance of small object detection and the training process.

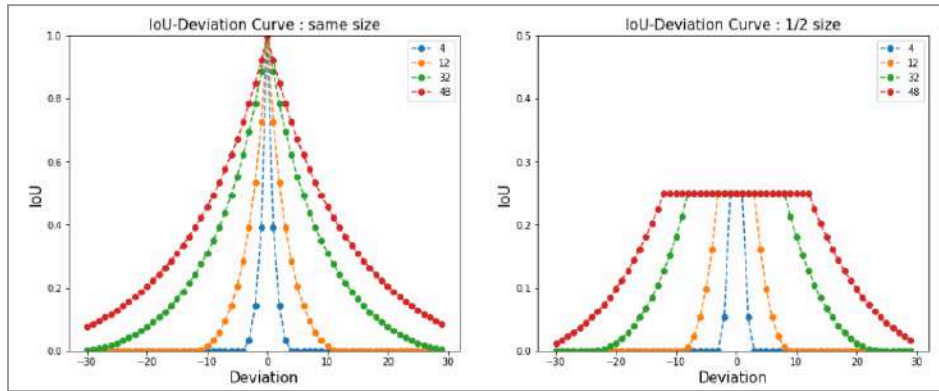


Figure 3.1: IoU-Deviation Curve

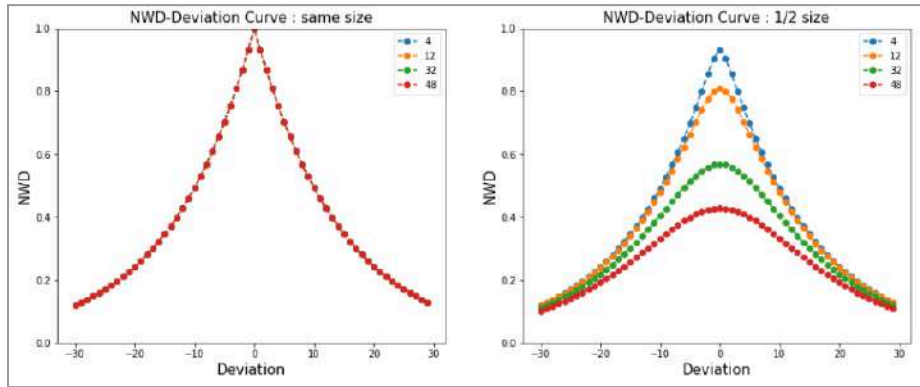


Figure 3.2: NWD-Deviation Curve

As seen in the above figures, every curve has a corresponding bounding box scale or size. Better localization accuracy is shown by a greater IoU, which means that the predicted bounding box closely matches the ground truth. The Wasserstein Distance, which calculates the "cost" of changing a distribution (bounding box) into another, is the source of NWD. A larger NWD denotes a lower degree of agreement between the anticipated and ground truth bounding boxes by suggesting a lower degree of similarity between the distributions.

Referring to the figure 3.1, flatter curves indicate that the model is robust, which is not observed in the case of NWD.

3.2 AU Drone Dataset Preparation

3.2.1 AU Drone Dataset

The dataset used in this research is based on aerial videos captured using a DJI Mavic 2 Pro drone at a multi-lane urban roundabout located in Ahmedabad city, Gujarat, India (latitude 23°03'21.2"N, longitude 72°30'21.6"E) [18]. The videos were recorded at a resolution of 3840 x 2160 pixels, with the drone positioned at a flight height of 80 to 85 metres to ensure visibility of all entry and exit points of the roundabout [18]. The drone's gimbal was set to 90° downwards to provide a top-down view of the traffic scene [18].

There are 5 videos from the AU Drone dataset that are utilised as a part of the research. The dataset consists of 17,079 video frames showcasing 1,251 objects and their movements. Table 3.1 offers a summary of the dataset and the frequency of each item category depicted in the five videos.

Statistics	DJI_009	DJI_010	DJI_011	DJI_012	DJI_021
Total Frames	3457	2352	2709	2075	6486
Total Tracks	213	150	204	171	513
Instances for each Object Class					
awning - tricycle	6188	3858	3237	2760	5455
motor	50230	35832	47792	37943	68954
truck	1757	1819	2340	3558	2638
pedestrian	5431	1125	4348	2176	6384
car	10413	10844	27460	18923	62595
bicycle	920	43	194	154	5515
people	945	31	20	8	148
van	634	455	1191	445	3135
tricycle	404	558	1644	1104	1875
bus	1	0	745	1069	4

Table 3.1: AU Drone Dataset Statistics

This dataset is a real-time dataset, gathered from a roundabout in an urban area of India and this serves as a great platform to study traffic infractions and patterns in developing nations. The elevated perspective of the drone offers a view of the roundabout facilitating an easier examination of vehicle behaviour, lane adherence and the interaction with road structures.

The traffic situation, in Ahmedabad, where the dataset is based, mirrors what you can expect to see in cities across India. These areas stand out for their mix of vehicles, heavy traffic flow and issues related to road conditions and adherence to traffic rules. Focusing on a lane roundabout the dataset offers a detailed look into a crucial type of intersection known for frequent traffic violations and congestion challenges.

3.2.2 Dataset Annotations

The AU Drone data consists of images capturing cars and people taken by drones. Each frame is tagged with bounding boxes. Annotations are provided on a frame, by frame basis in CSV files. The CSV file contains details, for each entry.

Attribute (as per dataset)	Detail
Frm	Represents the frame number for each image in the dataset.
Track	Indicates the unique tracking ID assigned to each object across multiple frames. It helps in identifying the same object in different frames.
Class	Specifies the class of the annotated object, such as car, bicycle or pedestrian.
xc, yc	Represent the x and y coordinates of the center point of the oriented bounding box.
w, h	Indicate the width and height of the oriented bounding box.
Direction	Represents the direction of the object
Rotation_Angle (deg)	Specifies the rotation angle of the bounding box in degrees. It represents the orientation of the object relative to the horizontal axis.
Velocity (kmph)	Indicates the velocity of the object.
SD (m)	Represents the distance of the object from the drone in

	metres. It helps to know the spatial context of the object.
BSZ_Ang (deg)	Indicates the angle of the object relative to the drone. It provides information about the viewing angle of the object.
Rotated_BB	Contains the coordinates of the rotated bounding box. It represents a set of four points in the order (x1, y1, x2, y2, x3, y3, x4, y4, x1, y1) defining the corners of the rectangle.
SD_poly_HML	Represents the distance polygon of the object from the drone or the camera sensor. It is expressed as a set of points defining the polygon.
BSZ_poly_L_HML	Indicates the left polygon of the object's bounding box.
BSZ_poly_R_HML	Indicates the right polygon of the object's bounding box.

Table 3.2: AU Drone Dataset Annotation Details

The annotations provide details about each object, including its category, location, direction, dimensions, speed and spatial relationships, with the drone or camera sensor. The additional features enable an analysis and understanding of how the objects behave in the drone footage while the oriented bounding boxes offer placement of each item.

3.2.3 Using and Testing with YOLO-World

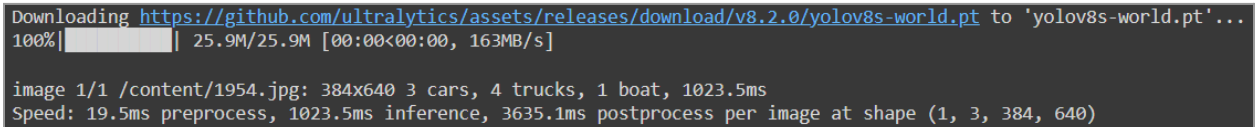
YOLO World is a user annotation tool that allows you to easily attach class labels and bounding boxes to images and videos. Its main purpose is to integrate with the YOLO (You Look Once) object detection system.

Extensive collections of data such as those used to recognize, link and associate images with text play a role in preparing YOLO World [14]. It offers an effective prompt-then-detect approach for inferring user vocabulary[14]. Because of this, YOLO-World is a flexible tool suitable for a wide range of vision-based applications. It can identify any object in a frame by using descriptive phrases[13].

Testing Results:

The results as seen in the figure 3.3 are not accurate. The image inputted has 48 objects in total including 3 awning tricycles, 1 bus, 14 cars, 26 motors, 1 pedestrian, 1 tricycle, and 2 trucks.

However the prediction shows 3 cars, 4 trucks, and 1 boat.



```
Downloading https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov8s-world.pt to 'yolov8s-world.pt'...
100%|██████████| 25.9M/25.9M [00:00<00:00, 163MB/s]

image 1/1 /content/1954.jpg: 384x640 3 cars, 4 trucks, 1 boat, 1023.5ms
Speed: 19.5ms preprocess, 1023.5ms inference, 3635.1ms postprocess per image at shape (1, 3, 384, 640)
```

Figure 3.3: YOLO-World Results

Thus, this model cannot be used for annotations as it can give inaccurate results, eventually leading to incorrect ground truth formation.

3.2.3 DOTA Dataset

DOTA (Database for Object deTecton in Aerial Images), is an extensive dataset used for object detection of aerial images[15]. It is useful for creating and assessing object detectors with both not tilted and oriented objects[15].

Key features:

- The dataset consists of images with different sizes that range from 800×800 to $20,000 \times 20,000$ pixels, taken from different sensors and platforms[16].
- It includes 18 categories with approximately 1.7 million Oriented Bounding Boxes[16].
- Comprises object detection on several scales[16].
- Experts use an arbitrary (8 d.o.f.) quadrilateral to annotate instances, catching objects of various sizes, shapes, and orientations[16].

Annotation format:

Each object in the dataset has an oriented bounding box (OBB) tagged on it. OBB is represented by $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, where (x_i, y_i) represents the i -th vertex of OBB[f]. The arrangement of the vertices is clockwise[f]. The annotations are visualised as follows. The starting point is represented by the yellow point, which can be found at (a) the top-left corner of a huge vehicle diamond, (b) the top-left corner of a plane, or (c) the centre of a baseball diamond[17].

In addition to OBB, every instance has a label that includes a category and a difficult value that indicates how difficult it is to identify the instance (1 for tough, 0 for not difficult)[f]. An annotation file with the same name as the image is saved[17].

```
x1, y1, x2, y2, x3, y3, x4, y4, category, difficult
x1, y1, x2, y2, x3, y3, x4, y4, category, difficult
...
```

Figure 3.4: DOTA Annotation Format (Src. github.io/DOTA/dataset.html)

3.2.4 Testing YOLOv8 with DOTA

Annotations are provided by the DOTA dataset in a particular format, usually as text files (see figure 3.4). Every annotation file is associated with a picture and holds details about the things that are visible in that image. The bounding box coordinates, other pertinent data, and the object category (class) are included in the annotation format.

For more understanding, the bounding boxes are drawn to the image frames, especially where the boxes are tilted. By using OpenCV, the annotations are read and the corresponding coordinates are extracted. The bounding boxes are then drawn on the images with their category. This visualisation of the annotations on DOTA images are seen in the figure 3.5.

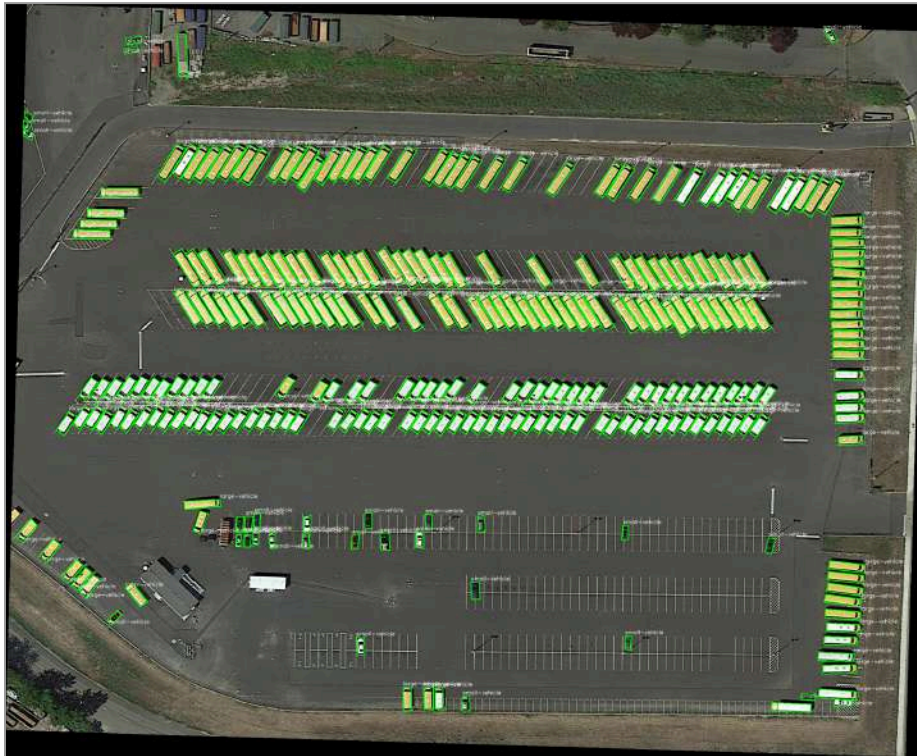


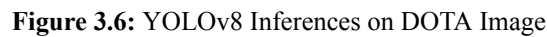
Figure 3.5: DOTA Annotations Drawn to Image

Testing YOLOv8:

YOLOv8, an advanced object detection model, builds on the achievements of earlier versions. Its object detection capabilities in frames and videos are intended to be quick, precise, and effective. Because it can recognise tilted items with oriented bounding boxes, the YOLOv8 is ideal.

Bounding boxes and class probabilities are directly predicted by YOLOv8 using a single neural network in a single forward pass. To manage objects of various sizes and aspect ratios, it makes use of strategies including object scaling, feature pyramids, and anchor boxes. To enhance the accuracy of bounding box regression and object recognition the model undergoes training that incorporates a mix of localization and classification losses.

Utilising a trained model named 'yolov8xobb.pt' for conducting inferences, on DOTA images yields the results displayed in Figure 3.6.



The results are promptly showcased in the environment for analysing predicted annotations. YOLOv8 provides adaptability by producing annotations in formats with popular options including;

- Page 23 of 47

- xyxy: This format represents the bounding box as the coordinates of the top-left (x1, y1) and bottom-right (x2, y2) corners.
- xyxyxyxy: This format represents the bounding box as the coordinates of the four corners (x1, y1, x2, y2, x3, y3, x4, y4) in a specific order.
- xyxyxyxyn: This is similar to xyxyxyxy, but with normalised coordinates (values between 0 and 1) and an additional class label (n).

In order to compare them with the DOTA annotations the projected annotations are displayed in the xyxyxyxy format. When comparing the projected annotations to the DOTA annotations we observe outcomes.

Based on these results, YOLOv8 model may be confidently used for object detection tasks on aerial photos, such the ones in the DOTA dataset.

3.2.4 Conversion of Annotations

The DOTA dataset works well with models as proven in previous tests using YOLOv8. It also supports oriented object detection (OOD) , an aspect of the AU Drone dataset.

However, the annotation of AU Drone dataset presents some critical challenges. As mentioned in section 3.2.2, the annotations of AU Drone are in CSV format, which differs from the DOTA format. To solve this problem, a conversion process is carried out to transform the AU Drone annotations to DOTA format.

The CSV annotations of the AU Drone dataset consists of a lot of attributes for every item, including the frame number, class label, width, height, centre coordinates, and rotational bounding box coordinates. The rotated bounding box coordinates are especially important for the conversion process, since they specify the coordinates of the oriented bounding box. Thus, the 'Rotated_BB' column was used and the coordinates were extracted from this column. The coordinates were converted in the DOTA format, which consists of the four corners of the bounding box (x1, y1, x2, y2, x3, y3, x4, y4) followed by the class label and a default difficulty score of 1.

Verification of the converted annotations: The annotations converted are then verified and compared with the original CSV annotations to confirm that there is no error during the process. The comparison can be seen in the figures 3.7 and 3.8. The conversion was initially carried out only on an ample number of annotation files. As the process is successful, annotations for all the frames of all 5 videos are converted to DOTA format.

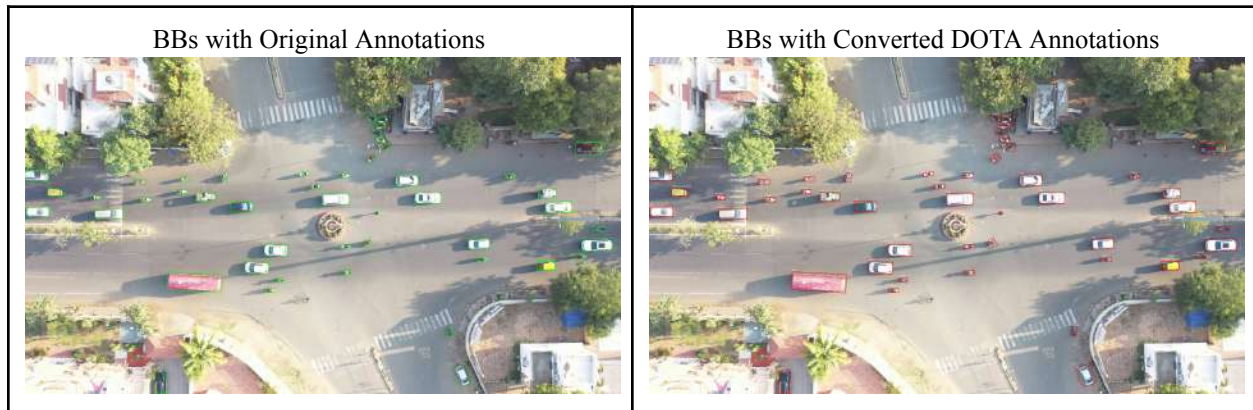


Figure 3.7: Verifying Converted DOTA Annotations

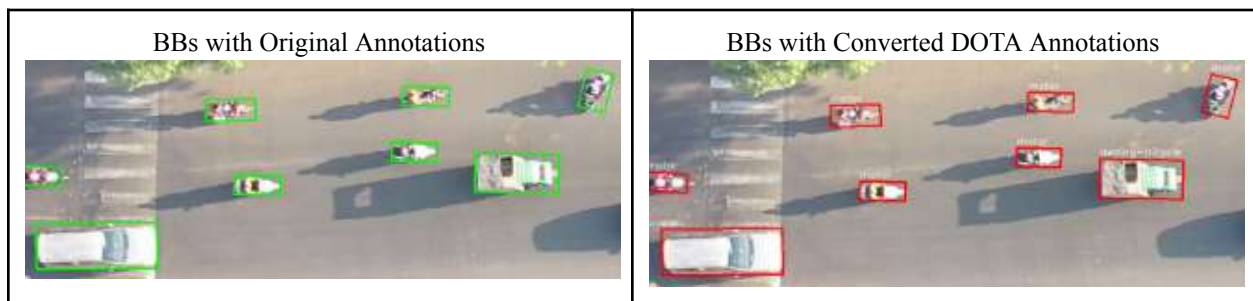


Figure 3.8: Verifying Converted DOTA Annotations

By this conversion, the AU Drone dataset becomes more compatible with the YOLOv8 and other oriented object detection (ODD) models.

3.3 Model Building

3.3.1 Box Boundary-Aware Vectors

The study posits the "Oriented Object Detection in Aerial Images with Box Boundary-Aware Vectors" as a new solution for location of oriented objects in aerial images using only the single-stage detector that is anchor-free called the BBA Vector model [6]. The model intends to rectify certain weaknesses of the conventional oriented methods and look to enhance the accuracy of detection of any rotated or unconstrained objects [6].

It develops the premise of keypoint-associated object recognition to deliver oriented bounding boxes (OBBs). The model is expected to predict both the centre of the objects as well as the boundary-aware vectors (BBAVectors) which will provide inbound boxes of the objects.

Model Architecture

In an U shaped vector BBA model the ResNet101 network serves as the core backbone network. Within this model an image is inputted to the backbone network. Processed to generate feature maps at different scales. The feature maps from the backbone are enhanced in size. The foundational features are combined with them using skip connections. This approach allows the model to incorporate both high level context and semantic significance well as lower level details and finer information.

The output of the model consists of four branches:

- Heatmap (P): Predicts the center points of objects.
- Offset (O): Refines the center point locations.
- Box Parameters (B): Regresses the BBAVectors (t, r, b, l) and the external size (w_e, h_e) of the OBBs.
- Orientation (α): Classifies the OBBs into horizontal bounding boxes (HBBs) or rotational bounding boxes (RBBs).

Figure 3.9 illustrates the architecture of the BBA Vector model.

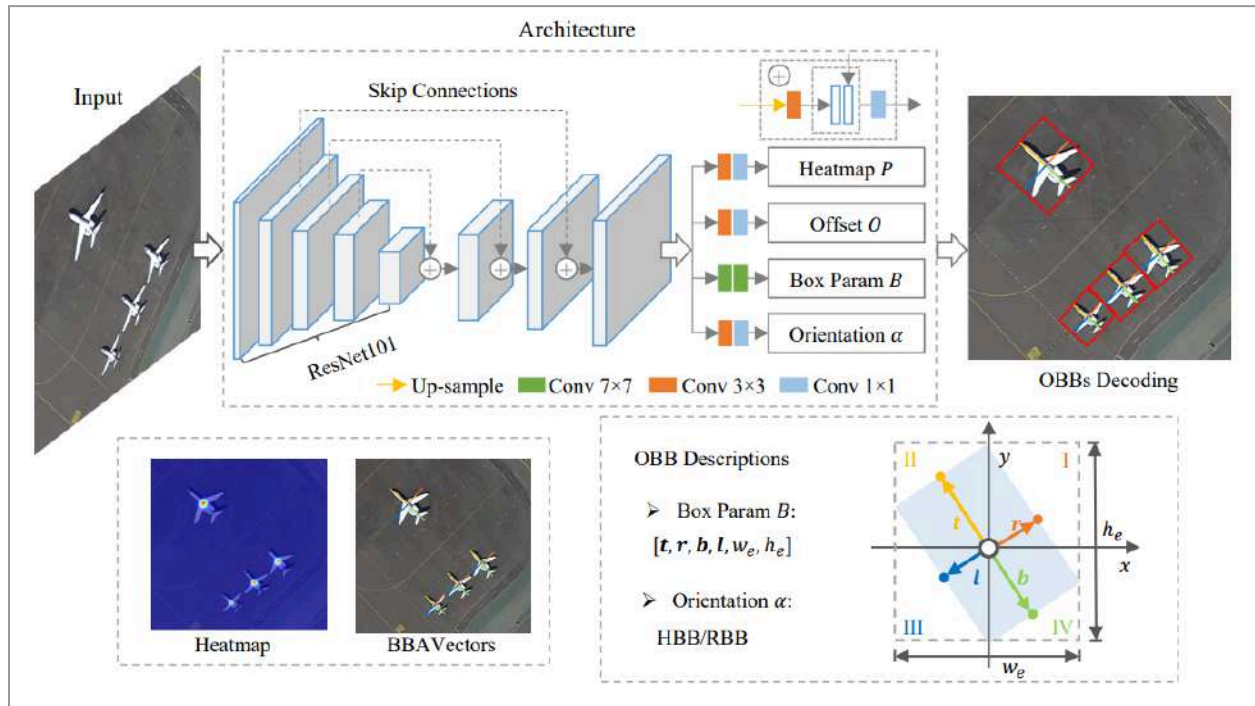


Figure 3.9: Architecture of BBA Vector Model

BBAVectors:

The BBA Vector model presents BBA Vectors (box vectors) as a method to depict oriented bounding boxes. Essentially BBA Vectors comprise four vectors (top, right, bottom left) that outline the boundaries of the OBB. These vectors are situated in the four quadrants of a coordinate system to cater to orientations.

Derived from the centres of entities, BBA Vectors provide versatility and creative liberty in contrast to parameters such as width, height and angle.

Figure 3.10 illustrates the architecture of the BBA Vector model.

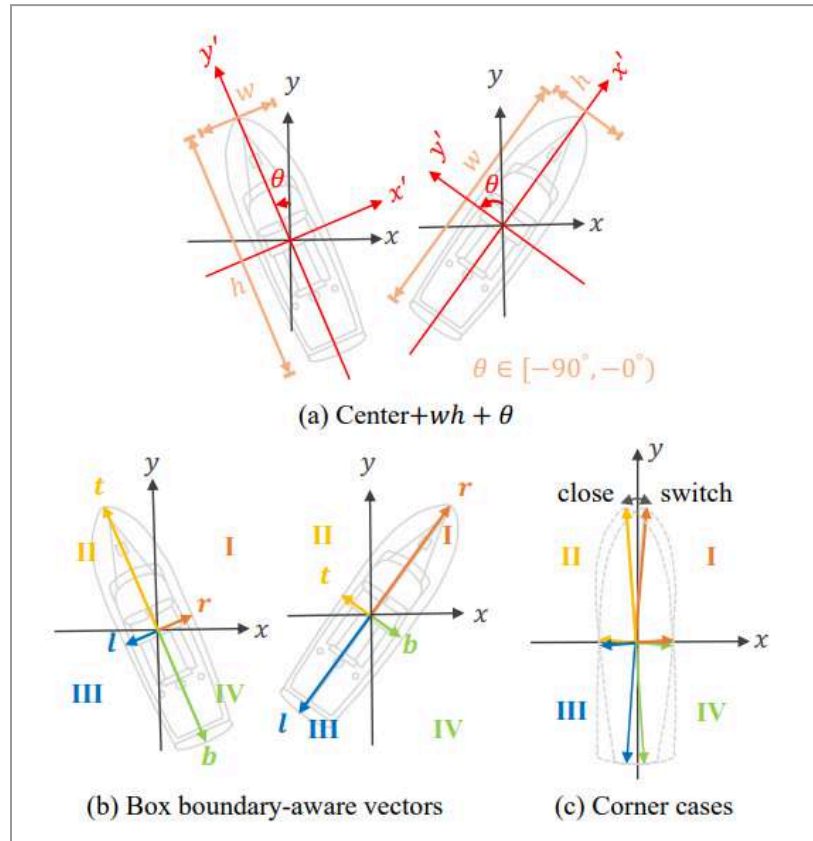


Figure 3.10: BBA Vector Method

In this plot, (a) is shortened as Centre+wh+ θ , which is the baseline method with parameters as w , h , θ being the OBB width, height, and angle. The w and h parameters of the OBBs are measured in the rotating coordinate systems, absolutely, for each object. (b) the proposed method where t , r , b , l are the Top, Right, Bottom and Left Box vectors representing the boundary-responsive. Quadrant-based boundaries are given by four vectors that are tied to the Cartesian system in any randomly given object. (c) obviously brings out the critical cases that are very near the xy -case.

Orientation Classification:

The BBA Vector model has added a branch to deal with the corner cases where the OBBs are almost on the xy planes. One of the branch predictors determines the probability of an object bounding box (OBB) being a rotational bounding box (RBB). The predicted probability if it is above the threshold (e.g., 0.5) then it is considered an RBB; otherwise, it is treated as an HBB.

The orientation classification assists the model to differentiate RBBs and HBBs and thereby capable of overcoming the challenging environment therefore accuracy is improved.

Loss Functions:

The BBA Vector model is trained using a combination of loss functions for each branch:

- Heatmap: Focal loss is used to handle the imbalance between positive and negative samples.
- Offset: Smooth L1 loss is used to regress the offset values.
- Box Parameters: Smooth L1 loss is used to regress the BBAVectors and external size.
- Orientation: Binary cross-entropy loss is used for orientation classification.

The total loss is a sum of the losses from each branch, guiding the model to learn the center points, BBAVectors, and orientation classes effectively.

Model Training

The AU Drone dataset images were resized to half their original size, and the annotations were adjusted accordingly. The model is optimized using the Adam optimiser with an initial learning rate of 1.25×10^{-4} . The total loss is a sum of the losses from each branch.

Hyperparameters: The model is trained on a 16 GB GPU system. The input images are resized to 1920 x 1080 pixels. The output feature maps have a resolution of 152 x 152 pixels.

The training is done on the frames which have approximately 15-20 objects per frame. From these 6000 bifurcated frames, 4000 frames are used for training the model.

Initially, the model is trained on 10 classes, but because there are relatively very few objects for some classes like buses, the number of classes are reduced by combining these classes together. Finally, the altered data with 7 classes – two-wheeler, three-wheeler, bicycle, pedestrian, people, cars, heavy vehicles, is trained. For better results, weights are added to different classes according to their variation.

3.3.2 YOLOv8

YOLO (You Only Look Once) is a popular family of real-time object detection algorithms known for their speed and accuracy. YOLO treats object detection as a regression problem, where it directly predicts bounding boxes and class probabilities in a single forward pass of the neural network. This approach eliminates the need for a complex pipeline and enables real-time performance.

YOLOv8 is the latest version of the YOLO architecture, released in 2023. It builds upon the success of its predecessors and introduces several improvements and optimizations. YOLOv8 offers enhanced performance, faster inference speeds, and greater flexibility compared to previous YOLO versions.

Model Architecture

Backbone Network: (Represented by layers P1 to P5 in Figure 3.11)

- YOLOv8 uses CNN (Convolutional Neural Network) backbone to extract features from the input image.
- The backbone network is pre-trained on a large scale dataset; it learns general image features from it.
- The most common architectures that are used in YOLOv8 include CSPDarknet, EfficientNet and ResNet.
- The spatial dimensions of the feature maps are reduced due to this backbone network. However, the number of channels are increased which captures both the low and high level features.

Neck Network: (Represented in Figure 3.11 by Upsample and Concatenate)

- The neck network aggregates and refines the features that are extracted by the backbone.
- The network consists of several layers that combine features from different scales and different resolutions.
- The feature maps from different levels of the backbone (P3 to P5) are upsampled and concatenated with the corresponding feature maps from the previous layer.
- PAFPN (Path Aggregation Feature Pyramid Network) is introduced in YOLOv8, which fuses features of different levels of backbone.
- The PAFPN enhances the multi-scale representation; also improves the detection of various sized objects.

Detection Head: (Represented in Figure 3.11 by conv and C2f layers)

- The anchor-free detection head is adopted by YOLOv8 which eliminates the need for predefined anchor boxes.
- The detection head is responsible for predicting the bounding boxes, class probabilities, and objectness scores.
- The conv layers are used to adapt the fused features from the neck network to the required dimensions for prediction.
- The C2f layers are the final convolutional layers (with 2x2 filter size) that generate the output predictions.
- It then predicts the coordinates of the bounding box, the probabilities and also the objectness scores for each of the cells in the feature maps.
- This approach simplifies the training process as it reduces the computational overhead that is associated with anchor box matching.

Loss Function:

- A combination of loss functions are employed to train the YOLO model effectively.
- It includes components for object classification, bounding box regression and prediction

of the objects.

- The Distribution Loss Function (DLF) is introduced which helps in handling the imbalance between foreground and background samples.
- The DLF improves the ability of the model to distinguish between objects and background.

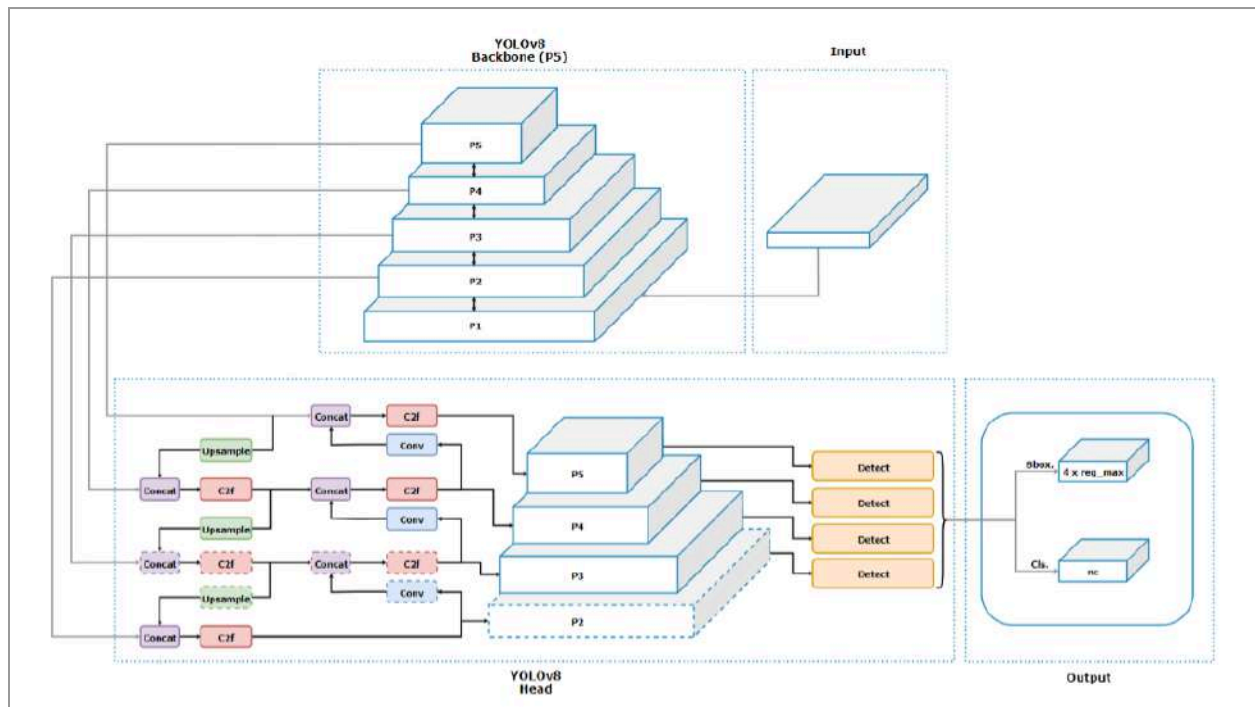


Figure 3.11: Architecture of YOLOv8

Model Training

The AU Drone Dataset is converted to DOTA format. But for training on YOLOv8, it is converted to YOLO OBB format.

The YOLO OBB format is an extension of the YOLO format to support the oriented bounding boxes. The YOLO OBB format can be seen in the following table:

class_id	x_center	y_center	width	height	angle
----------	----------	----------	-------	--------	-------

The angle in the table is the rotation angle of the bounding box in degrees, measured counterclockwise from the positive x-axis.

Once the annotations are converted, the dataset is organised in a specific directory structure which is compatible with YOLOv8.

A *.yaml* file, which is a configuration file, is created that specifies the data path, and model settings, including the class ids in it.

The model is trained on a system with a Quadro RTX 6000 GPU. The software environment includes NVIDIA-SMI. The CUDA version installed is 11.4. 15004 frames are used for training set, while 2075 frames are used for validation set. The training is performed for 100 epochs and a batch size of 4.

During the training, the model iterates over the training set and updates the parameters using backpropagation. This progress includes loss values and performance metrics. The best-performing model weights are saved for future uses.

Chapter 4

Results

4.1 BBA Vectors

In order to make the AU drone dataset consistent with the needs of the model, it is transformed into the DOTA dataset format. For a single epoch, a single video initially produced over 28,000 images, requiring a full day to process. The amount of objects per frame is determined by running statistics in order to optimise the training process. Approximately 6,000 frames are obtained by selecting frames containing 15–20 items. Approximately 4,000 of these 6,000 frames are used for testing, and the other frames are utilised for training.

The code is changed to allow for the ten classes that are present in the original dataset. By grouping related courses together, the number of classes is lowered to 7. This removes the issue of class imbalance.

The model achieved an overall accuracy of 84.71% on the dataset.

The detection speed of the model was 11 frames per second (FPS).

Table 4.1 in the report shows the detailed results for each class.

Class	mAP
Two-wheeler	97.18
Three-wheeler	94.07
Bicycle	95.66
Pedestrian	35.29
People	8.00
Cars	94.96
Heavy Vehicles	96.07

Table 4.1: BBA Vector Results (mAP)

The model performed exceptionally well for classes like Two-Wheeler, Three-Wheeler, Bicycle, Cars, and Heavy Vehicles, achieving accuracies above 94%. However, the model struggled with the Pedestrian and People classes, achieving accuracies of 35.29% and 8.00%, respectively.

The two images (Figure 4.1 and Figure 4.2) that visually demonstrate the model's performance. These images showcase the model's ability to detect and orient bounding boxes accurately around the objects of interest. The left side shows the original images with ground truth. The right side shows the predicted images with bounding boxes.



Figure 4.1: BBA Vector Results (Frame Comparison)

Upon analysing the results obtained from the Oriented Object Detection Model with Box Boundary-Aware Vectors, an orientation issue was observed for the 3-wheeler vehicles, specifically the awning tricycles. The model's predicted bounding box orientations for these vehicles were inconsistent with the expected orientations. However, upon further investigation, it was discovered that the ground truth annotations for the awning tricycles also exhibited incorrect orientations in some of the images, as shown in figure 4.2.

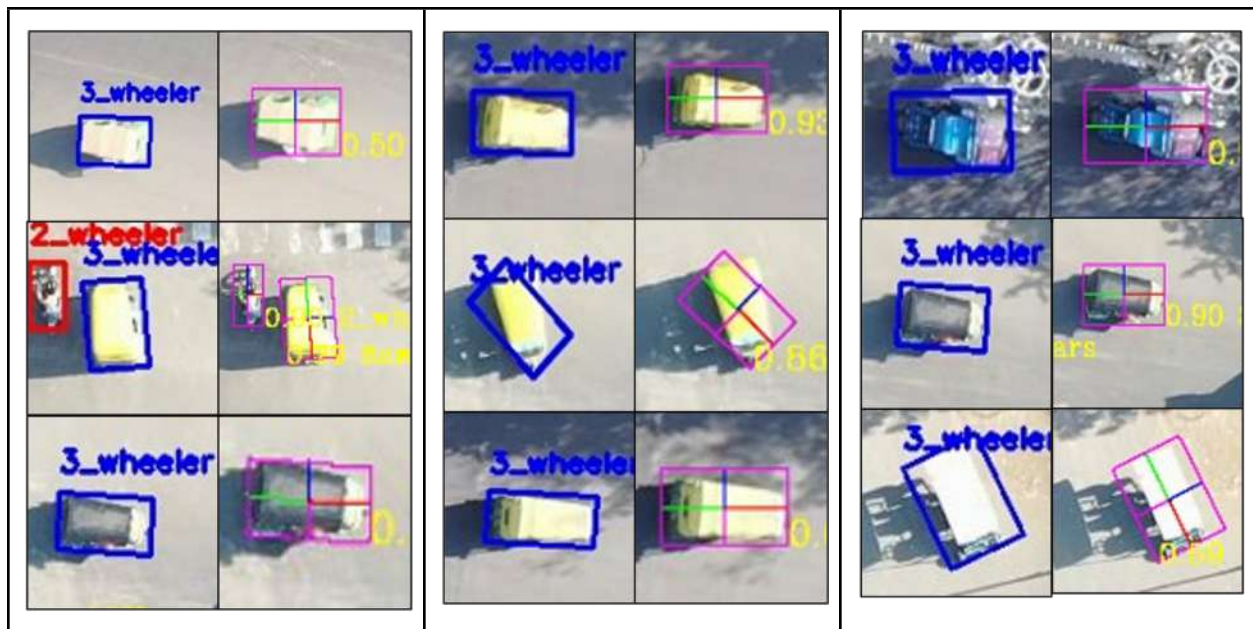


Figure 4.2: BBA Vector Results (3-Wheeler)

4.2 YOLOv8

The model was trained for 100 epochs, and the process was tracked during training. The below mentioned are the results obtained for box loss, classification loss, DFL loss, precision, recall and mAP. The batch predictions on the validation set are also seen below.



Figure 4.3: Validation Batch Labels



Figure 4.4: Validation Batch Predictions

The predictions from a batch of validation set are seen in the figure 4.4. The vehicles with the bounding boxes are correctly predicted. The bounding boxes are similar to the original labels(annotations).

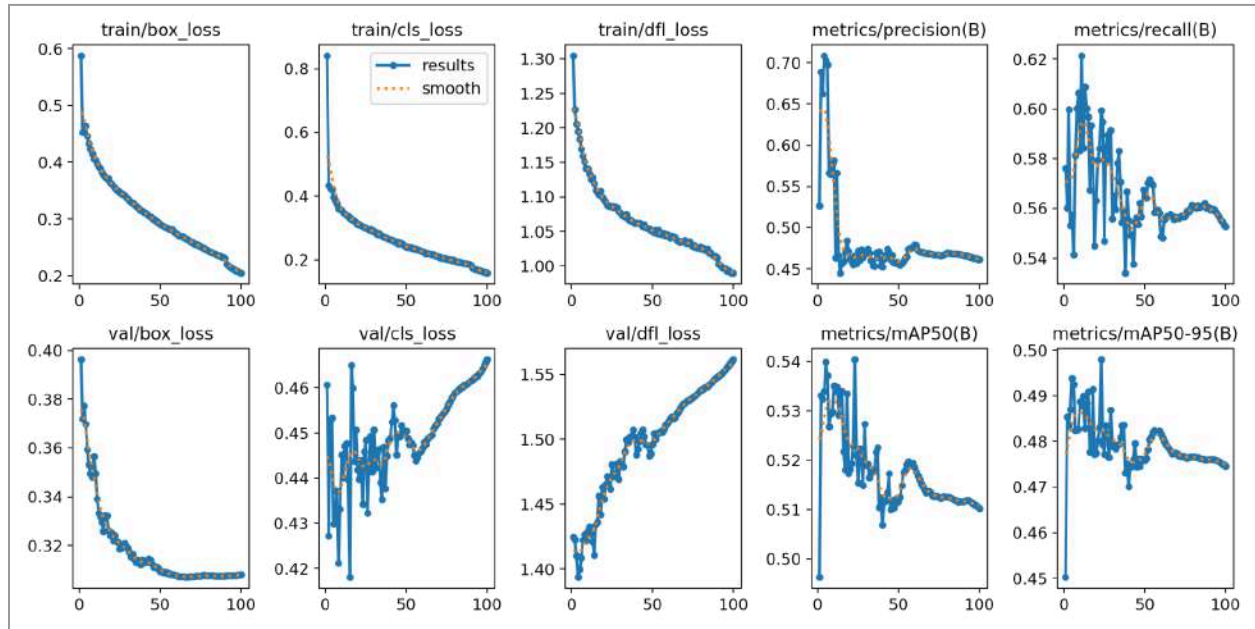


Figure 4.5: YOLOv8 Results (Metrics)

The training losses as seen in the figure 4.5, are decreasing over the epochs. This says that the model was learning and improving its ability to predict the bounding boxes, classify objects and refine the box predictions. The validation box loss also showed a decreasing trend, that shows that the model was generalising well to unseen data.

The precision and recall metrics remained relatively the same. The values are around 0.46 and 0.56 respectively. Thus, it can be said that the model maintained balance between correctly identifying positive instances and minimising false positives.

The mAP is shown in the figure 4.6. The mAP50, that is the mean Average Precision at IoU threshold of 0.5, remained around 0.51. This indicated that the model achieved an average precision of 51%. The mAP50-96 represents the average mAP from IoU thresholds from 0.5 to

0.95. This remained stable around 0.47. Thus, it can be concluded that the performance of the model is consistent for different IoU thresholds.

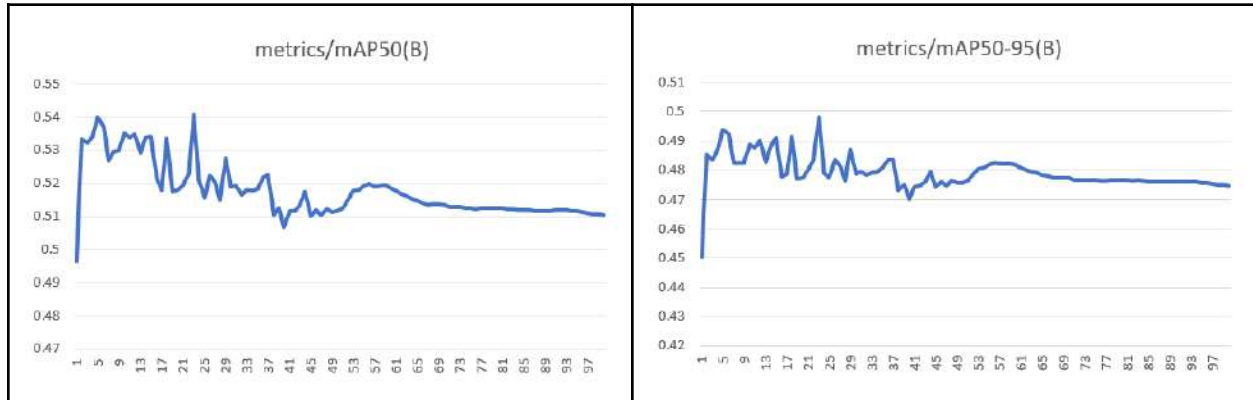


Figure 4.6: YOLOv8 Results (mAP)

Figure 4.7 is the confusion matrix of the results on all the classes. The diagonal entries show the correct classifications, while the off-diagonal entries indicate misclassifications. It is observed that the model performs well for some classes such as motors and cars. However, for other classes such as people, pedestrians and bicycles the model is struggling.

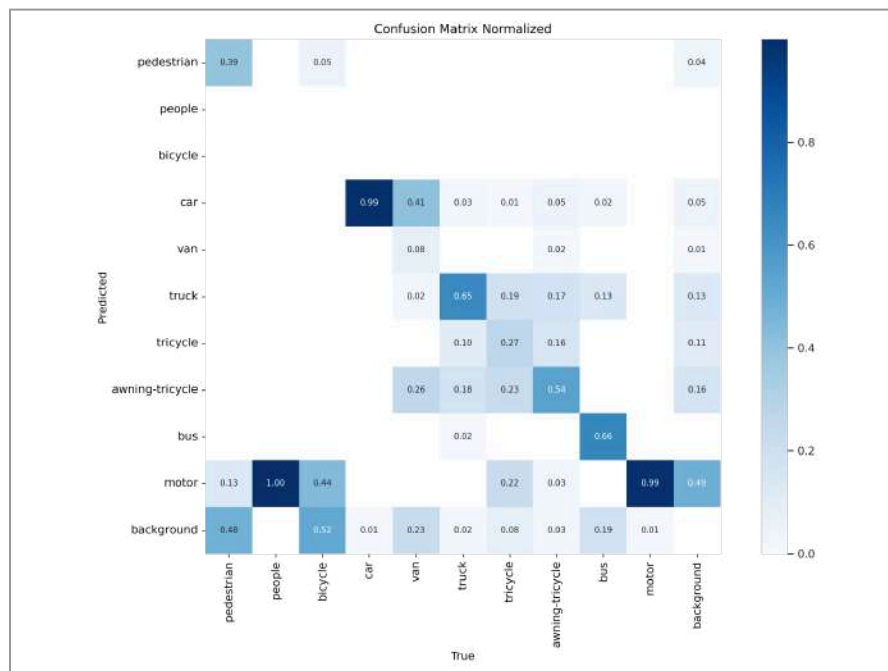


Figure 4.7: YOLOv8 Results (Confusion Matrix)

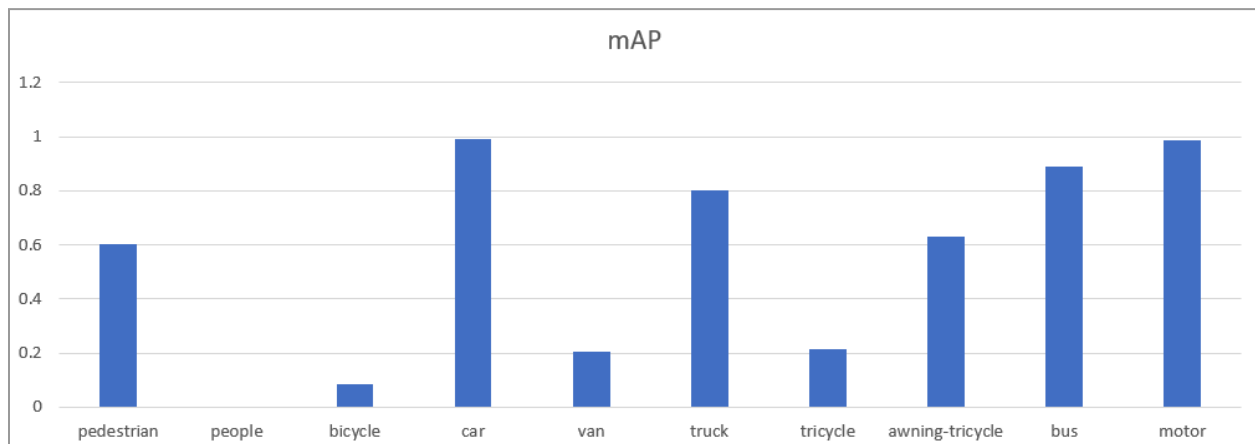


Figure 4.7: YOLOv8 Results (Confusion Matrix)

The mAP results across the classes shows that classes such like car, bus, motor and truck show good results. This is due to the number of instances for each of these classes. There are comparatively very less number of instances for people, bicycle, van and tricycle in the validation set, as well as the whole dataset. This results in lower accuracy of prediction of these classes.

Chapter 5

Conclusion

The AU Drone dataset has 10 different classes in the data. When the dataset is trained on any model, be it BBA Vectors or YOLOv8, shows similar results. The classes which have more number of instances train better as compared to those with lesser number of instances. This shows inconsistency among the number of classes in the dataset. Data augmentation is required to train the model perfectly.

Furthermore, the ground truth of the data has some issues. For example, the awning tricycle class has imprecise bounding boxes for some of the tracks. Thus, the predicted results also show the same inconsistency. This has to be corrected for proper ground-truth formation.

In conclusion, oriented bounding boxes are difficult to prepare and train the model on. The room for inaccuracy increases as the bounding boxes are oriented. More work needs to be done for the data preparation.

5.1 Project Outcomes

The project successfully demonstrated the application of UAV imagery and deep learning techniques for analysing Indian road traffic. The key outcomes of the project are as follows:

1. Creation of an AU Drone Dataset containing 17,079 video frames capturing the movement of 1,251 objects. This dataset acts as a foundation for studying traffic flow and vehicle interactions within roundabouts, in India.

2. Implementing OOD models like YOLOv8 and BBA Vectors for detecting oriented objects in aerial images. These models were trained on the AU Drone Dataset. Produced results in identifying and classifying relevant objects.
3. The BBA Vectors model achieved an accuracy of 84.71%, on the dataset, which was impressive given its performance in classes like Two wheeler, Bicycle, Cars, vehicles (all above 94%). However it highlights the importance of developing the model to account for missing classes such as pedestrians and people.
4. The project emphasised the importance of ground truth annotations. While reviewing the results of the BBA Vectors model discrepancies were found in the annotations, for 3 Wheeler vehicles (tricycles) specifically related to orientation errors. This underscores the need for annotation and quality assurance practices.
5. The project demonstrates the feasibility and potential of using UAV imagery and deep learning techniques for traffic analysis in the Indian context. The insights gained from this project can contribute to better understanding of traffic patterns, vehicle distributions, and potential traffic violations, aiding in effective traffic management and urban planning.

5.2 Real-World Applications

The outcomes of this project have significant real-world applications in the context of Indian road traffic management and urban planning. Some of the potential applications include:

1. Traffic Monitoring and Analysis: The system developed operates in time to track and analyse situations at busy intersections or crowded areas that may experience delays. For example by utilising drone footage traffic authorities can gather data on car volumes, traffic patterns, in specific areas and pinpoint potential congestion spots. This kind of analysis can enhance the effectiveness of data driven traffic control efforts to alleviate traffic congestion in those locations.

2. **Intelligent Transportation Systems:** The project can serve as a tool to assess the effectiveness of traffic systems and their impact, on both traffic flow and safety improvements. Real time object detection and classification enable functions such as automated traffic control, dynamic route planning and incident detection ultimately enhancing the quality of transportation networks and increasing safety levels.
3. **Urban Planning and Infrastructure Development:** The adjustment of traffic flow and traffic volume can serve as the foundation for projects such as establishing road networks, implementing traffic signal systems and allocating resources for infrastructure development. By considering the characteristics of road traffic experts can devise better strategies, for efficient and sustainable traffic management.
4. **Enforcement and Compliance Monitoring:** A sophisticated traffic management system can detect traffic violations automatically such as lane usage, unauthorised parking or driving in the wrong direction. By integrating this violation detection system authorities will be able to identify offenders and enforce penalties leading to improved adherence to traffic regulations and enhanced road safety.
5. **Accident and Incident Analysis:** When accidents or incidents occur the footage taken by drones, from above can offer information, for examining and investigating. By using object detection and tracking features it becomes possible to piece how things unfolded pinpoint factors that played a role and aid in making decisions based on evidence to prevent accidents and lessen their impact.

5.3 Future Work

Based on the project outcomes and observations, these is the future work to enhance the system and address the identified challenges:

1. **Improving Ground Truth Annotations:** As mentioned in the results, the ground truth annotations for the 3-Wheeler vehicles (awning tricycles) had some incorrect orientations. There is a need to revisit and refine the annotations for these specific vehicles. This can be done by carefully reviewing the video frames and manually correcting the orientations to ensure accurate and consistent ground truth data.
2. **Enhancing Pedestrian and People Detection:** Both the BBA Vectors model and the YOLOv8 struggled with the Pedestrian and People classes. There can be improvements in this area.
3. **Data Augmentation:** There is inconsistency in the number of instances in different classes, which is the cause of lesser accuracy. Thus, data can be augmented for these classes which can improve consistency of the dataset, eventually improving the accuracy of the models.
4. **Expanding the Dataset:** To improve the robustness and generalisation capabilities of the models, the Dataset can be expanded to include more traffic videos; maybe at different locations, and weather conditions. Collecting additional aerial footage from different urban areas in India can help capture the diversity.
5. **Real-Time Deployment and Optimization:** Future work can focus on optimising the models for real-time deployment, considering factors such as inference speed, computational efficiency, and resource constraints. Techniques like model compression, quantization, or edge computing can be explored to enable real-time processing of aerial footage on resource-constrained devices or in low-bandwidth environments.
6. **Integration with other Sources:** Integrating the aerial footage analysis with other data sources, such as traffic sensor data, GPS data, or weather information, can provide a more holistic understanding of traffic conditions.

Bibliography

- [1] “Papers with Code - Object Detection In Aerial Images,” *paperswithcode.com*.
<https://paperswithcode.com/task/object-detection-in-aerial-images>.
- [2] Viegas, Vanessa . “Why Is India’s Traffic Still among the Worst in the World?” Hindustan Times, 12 Mar. 2021,
www.hindustantimes.com/lifestyle/travel/why-is-india-s-traffic-still-among-the-worst-in-the-world-101615556859851.html.
- [3] Viegas, Vanessa. “Big Data Traffic Study Identifies India’s Fastest and Slowest Cities.” Haas News | Berkeley Haas, 5 Nov. 2018,
newsroom.haas.berkeley.edu/news-release/traffic-study-identifies-indias-fastest-slowest-and-most-congested-cities/
- [4] Singh, Meenu. “Revolutionizing City Transportation: The Influence of Artificial Intelligence on Traffic Management.” INDIAai, 18 Dec. 2023,
indiaai.gov.in/article/revolutionizing-city-transportation-the-influence-of-artificial-intelligence-on-traffic-management.
- [5] “India Traffic Report | TomTom Traffic Index.” India Traffic Report | TomTom Traffic Index,
www.tomtom.com/traffic-index/india-country-traffic/.
- [6] Yi, Jingru, et al. “Oriented Object Detection in Aerial Images with Box Boundary-Aware Vectors.” ArXiv (Cornell University), 1 Jan. 2020, <https://doi.org/10.48550/arxiv.2008.07043>.
- [7] “Build Software Better, Together.” GitHub, github.com/topics/oriented-object-detection
- [8] “Papers with Code - Oriented Object Detection.” Paperswithcode.com,
paperswithcode.com/task/oriented-object-detection.
- [9] “Papers with Code - an Overview of Object Detection Models.” Paperswithcode.com,
paperswithcode.com/methods/category/object-detection-models.
- [10] Yu, Xinyi, et al. “Oriented Object Detection in Aerial Images Based on Area Ratio of Parallelogram.” ArXiv (Cornell University), 1 Jan. 2021,
<https://doi.org/10.48550/arxiv.2109.10187>.

- [11] “V8.1.0 Release - YOLOv8 Oriented Bounding Boxes (OBB) · Ultralytics · Discussion #7472.” GitHub, github.com/orgs/ultralytics/discussions/7472.
- [12] Ultralytics. “VisDrone.” Docs.ultralytics.com, docs.ultralytics.com/datasets/detect/visdrone/.
- [13] ---. “YOLO-World (Real-Time Open-Vocabulary Object Detection).” Docs.ultralytics.com, docs.ultralytics.com/models/yolo-world/#train-usage.
- [14] “AILab-CVC/YOLO-World.” GitHub, 4 Mar. 2024, github.com/AILab-CVC/YOLO-World.
- [15] “Papers with Code - DOTA Dataset.” Paperswithcode.com, paperswithcode.com/dataset/dota.
- [16] Ultralytics. “DOTAv2.” Docs.ultralytics.com, docs.ultralytics.com/datasets/obb/dota-v2/.
- [17] “DOTA.” Captain-Whu.github.io, captain-whu.github.io/DOTA/dataset.html.
- [18] Bhavsar, Yagnik M, et al. “Vision-Based Investigation of Road Traffic and Violations at Urban Roundabout in India Using UAV Video: A Case Study.” Transportation Engineering, vol. 14, 1 Dec. 2023, pp. 100207–100207, <https://doi.org/10.1016/j.treng.2023.100207>.