Module Code: CSMAI

Assignment report Title: Group Coursework

Group member:
Kaiwen Gu 32830023
Zhixin Ding 32813906
Weijie Cui 32824514
Di Xiao 32817817

Date (when the work completed):08.05.2025

Actual hrs spent for the assignment: 80 hrs

Code: https://github.com/UoR-Vision/VisualDota

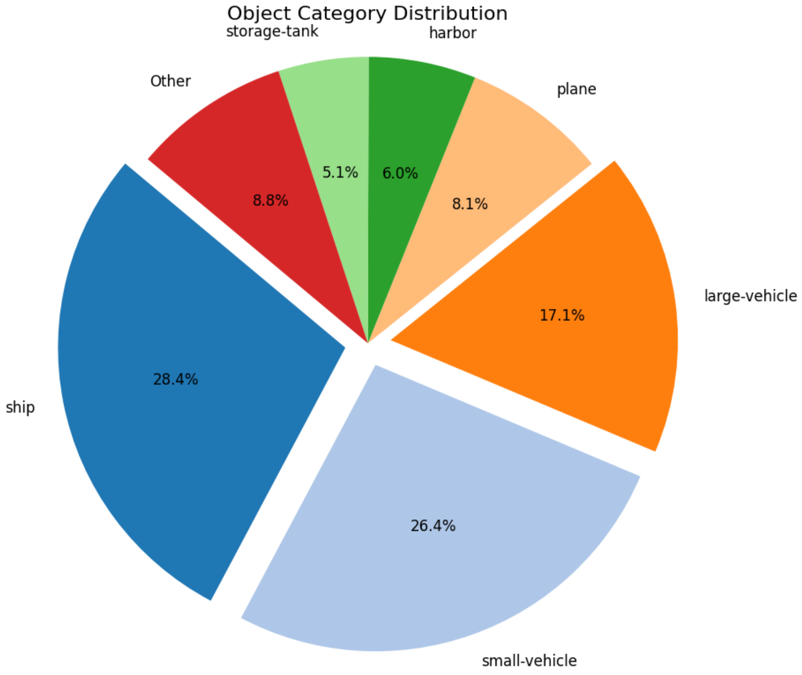
1 Introduction

The DOTA dataset is a large-scale benchmark designed for multi-object detection in aerial images. It contains 2,800+ high-resolution images (800×800 to 20,000×20,000 pixels) from multiple sources like Google Earth and satellites, with nearly 190,000 annotated object instances across 15 categories (e.g., plane, ship, vehicle, sports fields).Each object is labeled using oriented bounding boxes (OBBs) to handle complex orientations, and tagged by detection difficulty. This makes DOTA ideal for testing models in real-world scenarios involving scale, rotation, and dense object layouts.

In disaster response and rescue operations, rapid and accurate identification of critical infrastructure and affected zones is essential. Implementing multi-object detection models allows automated analysis of large-scale aerial imagery to locate damaged buildings, blocked roads, vehicles, and gathering areas for survivors. The DOTA dataset, with its extensive annotations of structures such as ships, vehicles, and sports fields, offers a practical foundation for training AI systems aimed at enhancing situational awareness during emergencies. Leveraging advanced deep learning detectors trained on this dataset can significantly prioritize rescue efforts, and improve coordination in high-pressure rescue scenarios.

2 Exploratory data analysis

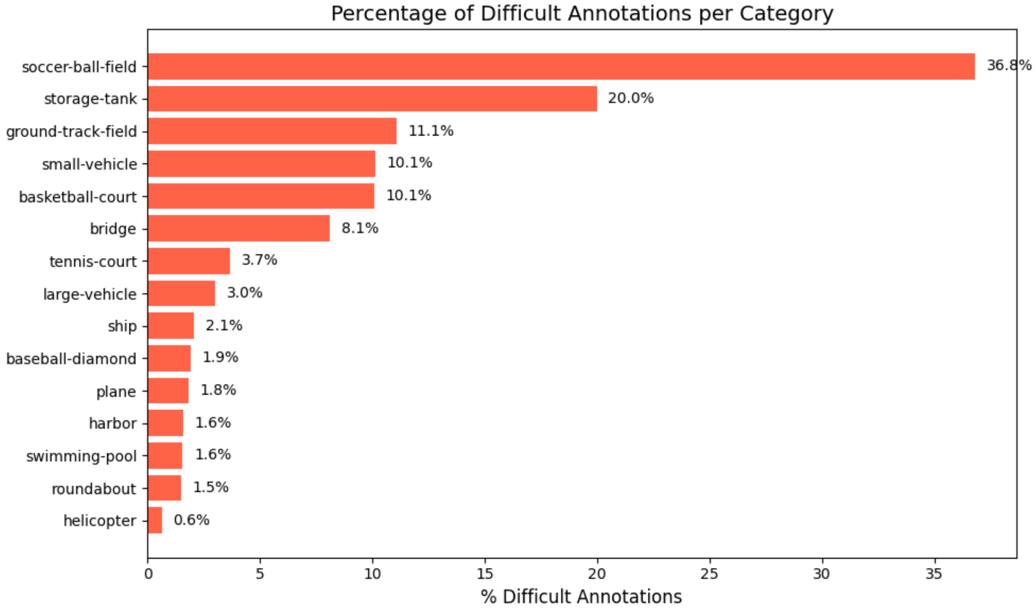
2.1 Object Category Distribution



The pie chart visualization revealed that “ship,” “small-vehicle,” and “large-vehicle” categories dominate the dataset. These three classes alone account for over 70 percent of all labeled instances, highlighting a clear imbalance in class representation. The remaining categories, such as “storage-tank” and “harbor,” contribute significantly less, and all other minority categories were grouped under “Other.” This indicates a long-tailed distribution that may pose challenges during model training.

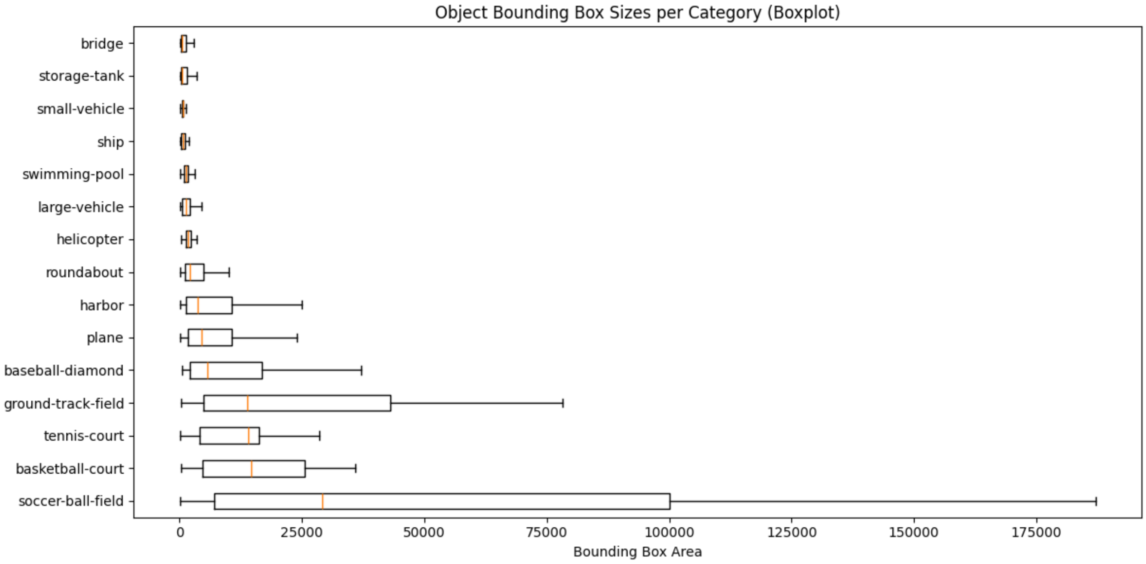
From a modeling perspective, this imbalance implies that detection models may perform disproportionately well on dominant categories while struggling with underrepresented ones. Addressing this requires techniques such as class weighting, resampling, or augmentation to ensure balanced learning outcomes.

2.2 Percentage of Difficult Annotations Per Category



The bar chart visualizes the percentage of instances per category that are labeled as difficult. This label is assigned when an object is challenging to identify due to factors like low resolution complex background or occlusion. Soccer-ball-field and storage-tank are the most challenging with difficulty rates of 36.8 percent and 20.0 percent likely due to visual clutter and structural ambiguity. Ground-track-field and small-vehicles also exceed 10 percent indicating moderate complexity. In contrast categories like ship, plane and helicopter are easier to detect with difficulty rates below 3 percent. These findings emphasize the need for class-specific strategies such as data augmentation and adjusted detection thresholds to ensure balanced model performance across all categories.

2.3 Object Bounding Box Sizes Per Category



The box plot illustrates the distribution of bounding box areas for each object category in the DOTA dataset. The soccer-ball-field and ground-track-field exhibit the largest bounding box areas with extremely wide ranges. These objects often occupy large portions of aerial images and are easier to detect due to their scale and prominence. In contrast objects like small-vehicle storage-tank and helicopter show very small bounding box areas with tight distributions. These smaller objects are more susceptible to being overlooked especially in high-density scenes or lower-resolution inputs. Mid-range categories like basketball-court baseball-diamond and harbor exhibit moderate variation with a mix of easily and moderately challenging instances. These distributions can anticipate which object types may require more augmentation or resolution adjustments during training.

3 Data pre-processing

3.1 YOLO v1

To prepare the DOTA dataset for training with YOLOv1, image and label data were first collected and organized into structured directories. The preprocessing pipeline includes loading each image from disk, resizing it to a fixed resolution of 448x448 pixels, and converting its color space from BGR to RGB. Images were normalized to a [0,1] range by dividing pixel values by 255 and converted into PyTorch tensors. For the labels, corresponding .txt annotation files were read and parsed to extract object class and bounding box parameters. These annotations were transformed into a tensor format compatible with YOLOv1's output, which encodes class, objectness confidence, and bounding box coordinates for each grid cell. This process ensures co nsistent input-output structure and supports mini-batch training by stacking the transformed images and encoded labels into a PyTorch DataLoader.

3.2 YOLOv11n

Due to the high resolution of DOTA images, it is difficult to use the original images. So, the official toolbox provided by the DOTA website is used to split the images into smaller patches suitable for model input, with an appropriate gap to preserve targets near the edges. The corresponding annotation files have also been adapted to the split images. In addition, the annotation format was converted into a format recognisable by YOLO. The final dataset structure is shown below and can be automatically read by the Ultralytics framework.

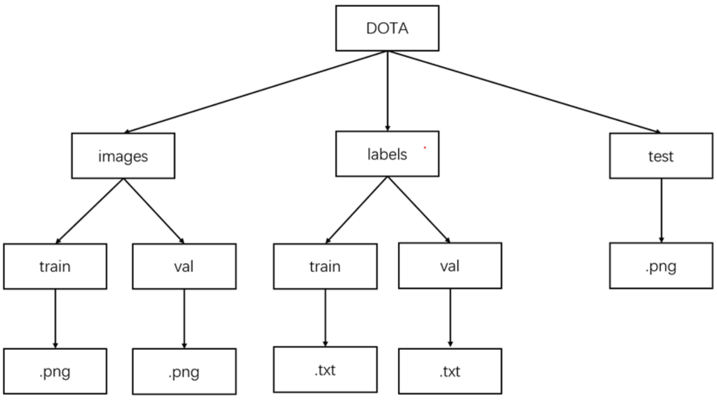


Figure dataset structure

3.3 Faster R-CNN

Similar to YOLOs, to fit the high resolution remote sensing dataset, images were cutted into 5*5 image tiles. And then they were normalized and resized to min_size = 800 and max_size = 1333 for prediction, which is the model preset processing.

4 Machine learning model

4.1 YOLOv1

4.1.1 Summary

YOLOv1 (You Only Look Once) is well-suited for the DOTA dataset because it treats object detection as a single regression problem (Jain, 2025). Instead of using region proposals and classification separately like traditional methods, YOLOv1 processes the entire image in one pass, allowing it to be both fast and efficient. YOLOv1 divides the image into a 7x7 grid and predicts multiple bounding boxes and class probabilities for each cell, which works well for detecting small and overlapping objects. Its speed advant age also supports near real-time applications like disaster response and situational awareness, where quick decision-making based on object detection results is crucial.

4.1.2 Model Architecture and Loss Function

The YOLOv1 model includes two main parts: a convolutional feature extractor and a fully connected prediction head. The feature extractor consists of a series of convolutional blocks that gradually reduce spatial resolution while capturing higher-level features. The head flattens these features and passes them through two fully connected layers, finally outputting predictions for each cell in the 7x7 grid.

The model uses a custom loss function with four parts: coordinate loss for accurate box positioning, objectness loss to boost confidence for real objects, no-object loss to reduce false positives, and class loss for correct classification. Two hyperparameters, λ_coord and λ_noobj, balance the impact of localization errors and empty grid cell penalties, helping YOLOv1 learn localization and classification together effectively.

4.1.3 Result and Discussion

The YOLOv1 model was trained on the DOTA dataset for 10 epochs. The training loss decreased steadily from an initial value of 11.82 to 0.158, indicating that the model successfully learned to detect and classify objects within the aerial images. This reduction in loss reflects improved accuracy in predicting bounding boxes and object classes over time.

However, there are several limitations observed. First, the model’s use of axis-aligned bounding boxes prevents it from accurately representing rotated objects, which are common in aerial imagery such as tilted planes or diagonal ships. Additionally, YOLOv1’s fixed grid structure leads to poor performance on small or densely packed objects, as multiple objects falling within a single grid cell cannot be separately detected.

4.2 YOLOv11n

4.2.1 Summary

YOLOv11n is the smallest and fastest version of YOLOv11, making it ideal for use in resource-constrained environments such as mobile and embedded systems. Although it has fewer parameters than the YOLOv11-L and YOLOv11-X versions, it maintains strong performance in small object detection and real-time applications. These characteristics make it particularly well suited to detecting the densely packed objects in the DOTA dataset.

4.2.2 Model Setup and Training

The model used in this part is YOLOv11n-OBB, a YOLO variant that supports Oriented Bounding Boxes (OBB). This model is based on the YOLOv11n architecture but introduces an angle regression branch to handle rotated object detection.

To improve the detection of small objects in remote sensing images, the .load() method to load pre-trained COCO weights. Since most network layers have the same structure, the model can automatically load matching weights, while mismatching parts are automatically initialised. Although the ideal training input size is 1024 × 1024, we use 768 × 768 for training due to GPU memory limitations. While this results in some performance loss, the impact is acceptable. In addition, in small sample remote sensing scenarios, transfer learning from COCO significantly accelerates model convergence and improves initial accuracy, making it an effective strategy.

4.2.3 Result

The figure shows the evolution of the model's performance over the 12 training rounds. The training loss (box loss, cls loss, dfl loss) and validation loss as a whole continue to decrease, indicating that the model is converging steadily; among the accuracy indicators, Precision reaches the highest of 0.62, Recall reaches 0.56, mAP@0.5 finally reaches 0.57, and mAP@0.5:0.95 reaches 0.38, all of which show a steady increasing trend. The results validate the performance of the model in rotating the target. These results confirm the effectiveness of the model in the rotating target detection task.

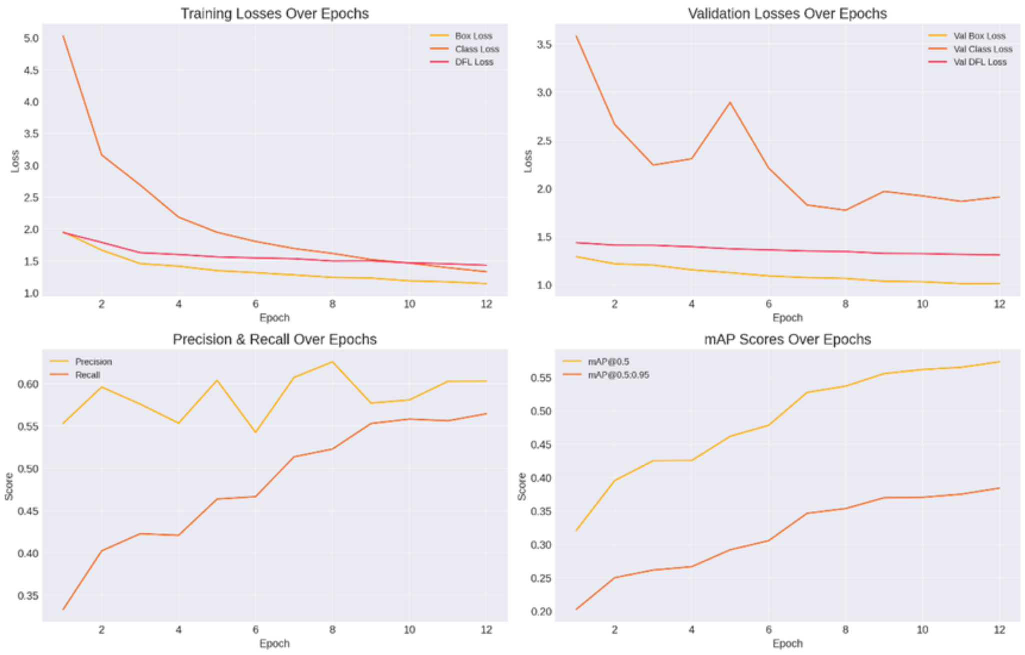


Figure Training Performance Metrics Over Epochs

4.3 Faster R-CNN + Transfer Learning

4.3.1 Summary

Faster R-CNN is an advanced object detection model based on region proposal algorithms(S. Ren, 2017). It introduced a Region Proposal Network (RPN), which enhances full-image convolutional features for better performance. The RPN is trained to provide regional proposals. The base model used in this project was trained to fit on PASCAL VOC and MS COCO datasets.

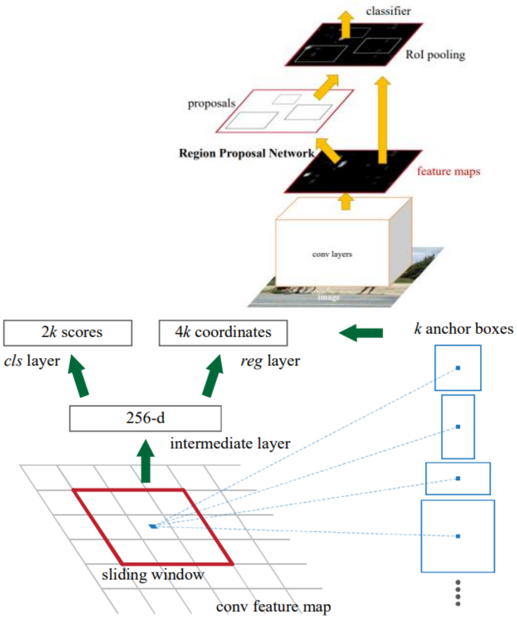


Figure: **Left**: Faster R-CNN. **Right**: Region Proposal Network (RPN).

4.3.2 Justification of Faster R-CNN

To adapt to the DOTA dataset, we used Transfer Learning. The top layer of the model was changed to 15 categories, and the layer was unfrozen and retrained. After several training attempts, the model failed to find any objects. After comparison, we found that models based on low resolution images (COCO) are not suitable for prediction on high resolution images (DOTA). To address this issue, we cut the image into 5*5 image tiles for prediction, and then merge the prediction results into the final output.

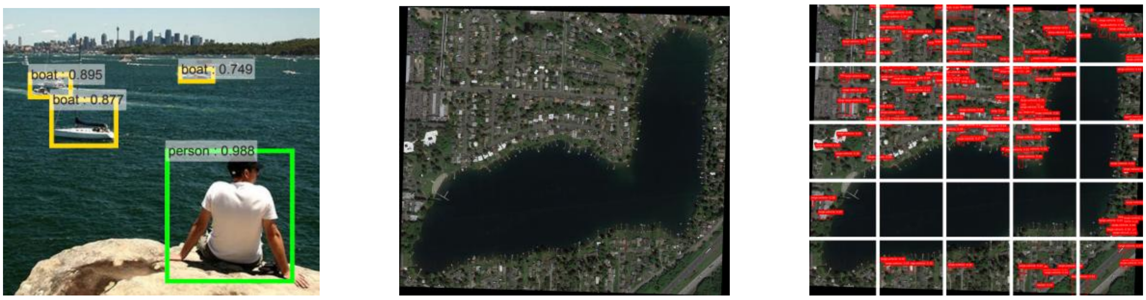


Image: Left: selected example of COCO. Middle: selected example of DOTA. Right: selected results of 5*5 image tiles solution of DOTA.

4.3.3 Results and Discussion

The figure shows the change of the loss of training and validation datasets in 12 training rounds. In the first 6 rounds, both the training loss and the validation loss decreased rapidly, and then entered a state of gentle fluctuation near 0.6 and 0.9, respectively. The loss consists of location loss, object loss and type recognition loss, which leads to the uncertainty of gradient descent. In addition, for the high-density DOTA dataset, the horizontal bounding box (HBB) causes the target and the box to overlap, which seriously affects the accuracy of the loss calculation.

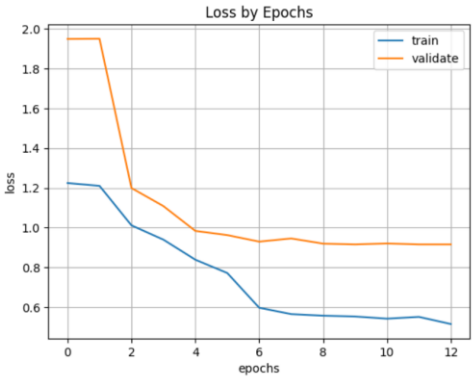
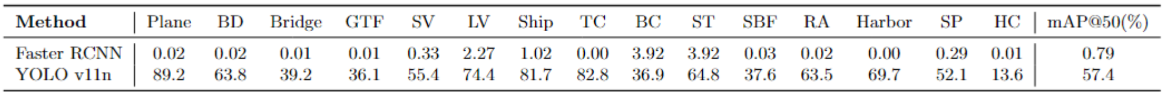


Figure: Training Loss Over Epochs

5 Evaluation and Comparison across the Models Built

5.1 Performance Measures

The objective of this analysis was to evaluate and compare object detection models on the DOTA dataset using mAP at IoU threshold 0.5 as the primary performance metric. The mean Average Precision is widely adopted in remote sensing tasks and reflects the area under the Precision Recall Curve(Han, Ding, Xue, & Xia, 2021). YOLO v1 is the original version of the YOLO series and is limited by its use of a fixed 7×7 grid and axis-aligned bounding boxes which makes accurate mAP calculation difficult. It struggles with small or overlapping objects and cannot handle rotated bounding boxes to compare with YOLO v11n.



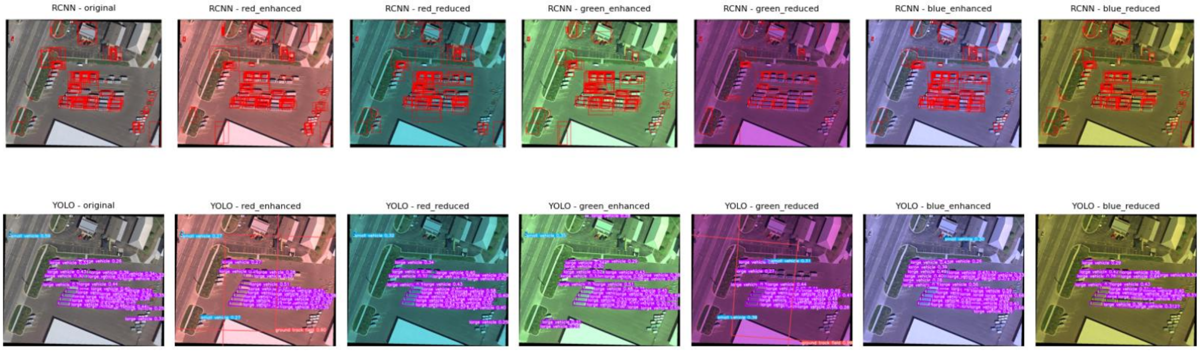
Method	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP@50(%)
Faster RCNN	0.02	0.02	0.01	0.01	0.33	2.27	1.02	0.00	3.92	3.92	0.03	0.02	0.00	0.29	0.01	0.79
YOLO v11n	89.2	63.8	39.2	36.1	55.4	74.4	81.7	82.8	36.9	64.8	37.6	63.5	69.7	52.1	13.6	57.4

mAP comparison

Foremost, compared to other models, YOLO v11n uses a more mature pre-trained model, which significantly reduces the risk of overfitting and provides strong robustness. Second, unlike Faster R-CNN, which does not support rotated bounding boxes, YOLO’s current version accommodates them. This improves detection accuracy and the ability to recognize complex objects, especially in structures like bridges where objects are elongated and arbitrarily oriented. Even with K-means redesigned anchor boxes and rotation augmentation, Faster R-CNN struggles with alignment and accuracy. Lastly, model evolution plays a role—Faster R-CNN is relatively outdated, while YOLO continues to be optimized.

5.2 Comparison of Color Channels

In this step, a set of images was selected and the RGB color channel was increased and decreased by 50%. After that, the enhanced images and the original images were used to predict. Here are some of the results.



color comparison

Overall, the reduction of the green channel led to a noticeable tendency for models to incorrectly classify areas as ground track fields, which indicated a strong dependency on the green channel information for determining the ground track fields. Also, the experiment showed that adjusting the blue channel had the tiniest impact on the confidence score of categories such as airplanes and cars. This suggests that the blue channel is no longer an influence for these categories. In contrast, the red channel significantly affected the confidence score across several categories, clarifying the high importance of the red channel for most categories.

Additionally, an attempt was made to adjust the image brightness. Unlike the significant changes in the channel, brightness had almost no effect on the prediction results. After analysis, it appears that in the context of object detection tasks, the model puts greater power on the shape of the objects rather than brightness

.

5.3 Future work

Due to time and computational resource limitations, some planned tasks were not fully implemented. For the Faster R-CNN model, only the RoI Heads layer was fine-tuned during training. Full parameter training is required to adapt the DOTA dataset. Additionally, anchor boxes were not optimized. Although k-means clustering was used to divide the labels into nine sizes and calculate appropriate anchor box sizes, modifying the anchor boxes requires retraining the model. This adjustment was not made due to time limitations. In future research, we plan to optimize the anchor box design in Faster R-CNN, retrain the model, and evaluate its impact on performance.

Furthermore, in terms of data augmentation, Faster R-CNN does not support rotated bounding boxes, and the dataset is relatively small, so data augmentation was not applied. This limitation led to suboptimal performance when the model encountered more complex objects, such as bridges. Future work will explore effective methods for implementing rotation augmentation in Faster R-CNN to improve the model’ s ability, especially in detecting complex objects.

Regarding to the YOLOv11n model, currently, the training process mainly involved model weights fine-tuning. Several training strategies and data augmentation techniques were applied by default during the training stage. In future work, more advanced customized training strategies, augmentation methods and remote sensing datasets are suggested to improve the model's performance and generalization capabilities.

6 Conclusion

This study evaluated multiple deep learning models for multi-object detection using the DOTA dataset with a focus on measuring model performance through mAP at IoU threshold 0.5. The analysis revealed that YOLOv1 achieved basic object detection capability but was limited by its fixed grid structure and inability to process rotated bounding boxes. As a result, it struggled with high-density scenes and irregular object orientations which are common in remote sensing imagery.

YOLOv11n demonstrated superior performance by supporting oriented bounding boxes and leveraging transfer learning from COCO weights. It achieved strong results in precision recall and mAP while maintaining efficiency in training. In contrast, Faster R-CNN showed reduced performance due to its reliance on horizontal bounding boxes and limited adaptability to complex spatial arrangements. The comparison confirmed that model architecture pre-training strategy and geometric support play critical roles in detection accuracy.

Overall the results highlight the importance of selecting models that align with dataset characteristics such as object orientation and density. Future improvements should focus on rotation-aware data augmentation anchor box optimization and full fine-tuning to further enhance model robustness and accuracy in remote sensing multiple object detection tasks.

Group Reflection

Name	Clearly specify your individual contributions in the required tasks	Please confirm your expected level of effort for the assignment	Signature
Kaiwen Gu	1. Dataset selection & preprocessing	100	
	2. Modelling	100	
	3. Evaluation	100	
	4. Coding	100	
	5. Report writing	100	
Zhixin Ding	1. Dataset selection & preprocessing	100	丁治鑫
	2. Modelling	100	
	3. Evaluation	100	
	4. Coding	100	
	5. Report writing	100	
Weijie Cui	1. Dataset selection & preprocessing	100	WeijieCui
	2. Modelling	100	
	3. Evaluation	100	
	4. Coding	100	
	5. Report writing	100	

Di Xiao	1. Dataset selection & preprocessing	100	肖迪
	2. Modelling	100	
	3. Evaluation	100	
	4. Coding	100	
	5. Report writing	100	

Individual Reflection

Kaiwen Gu:

In this project, my primary role involved conducting exploratory data analysis (EDA) and developing the initial YOLOv1 model pipeline. From there, I implemented the YOLOv1 training pipeline, configured the custom loss function, and managed training iterations while tracking loss convergence to ensure the model was learning effectively.

Beyond the initial development, I actively contributed to evaluating YOLOv1’ s limitations and discussed selecting better-suited models with the team. I also discuss with the team on the evaluation process, including setting up metrics such as mAP and comparing model performance. This collaborative and iterative process allowed our team to identify the most suitable model architecture for high-accuracy detection in aerial imagery.

Zhixin Ding:

In this project, I mainly focus on the model construction of YOLOv11n and the data preprocessing part.After reading related papers, I found that the pre-training model can significantly improve the effectiveness of the model, so I chose to use a pre-training model as the basis of migration learning to train the DOTA dataset.The parameters of the model are initialised by using the pre-training weights on the COCO dataset, and then the model is fine-tuned on the target dataset, which effectively reduces the training time, avoids overfitting, and improves the generalisation ability.I also discussed with the team the differentiation shown by the model in recognising different objects, the reasons for this differentiation, and where it can be optimised in the future.

Weijie Cui:

My main responsibilities in this project are sorting out and selecting models for the DOTA dataset and building Faster R-CNN trained model. I researched some multi-object detection and image segmentation algorithms, and shared my ideas to my teammates, then finally, we decided 3 algorithms: building a YOLO V1 model from scratch, a trained YOLO model and a trained Faster R-CNN model. When I adapted the Faster R-CNN model to the DOTA dataset, I developed some image cutting methods to achieve improved recognition accuracy with minimal computing resources.

Additionally, my contributions to project coordination included managing GitHub, Adapting programs to local and Colab dual development environme nts, which effectively facilitates team collaboration.

Di Xiao:

In this project, my main duty was model evaluation. I reviewed extensive literature and papers, selecting mAP as the sole metric. Regarding the mAP calculation, I found that the official tool provided by DOTA was obsolete and conflicting with Python 3. Therefore, I created a new method for calculating mAP to ensure compatibility with the current working environment. Also, I took on other code development tasks related to the evaluation process. About working with the HBB models, I operated the K-means clustering to find a suitable size of the anchor boxes. Additionally, I also made efforts in fine-tuning various models.

Reference

Han, J., Ding, J., Xue, N., & Xia, G.-S. (2021). ReDet: A rotation-equivariant detector for aerial object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2786–2795). IEEE.

Ren, S., He, K., Girshick, R., & Sun, J. (2017b). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi.org/10.1109/tpami.2016.2577031

Jain, A. (2025, April 15). Deep dive into YOLOv1 - Abhishek Jain - Medium. Medium. https://medium.com/%40abhishekjainindore24/deep-dive-into-yolov1-c70111debe60