# ENHANCED OBJECT DETECTION ON AERIAL CITYSCAPES VIA AUGMENTATION AND YOLO VARIANTS

*Vu Minh Hieu, Nguyen Trong Duc, Do Thanh Ha, Du Duc Tien, Phan The Son*

mhieu060@gmail.com, nguyentrongduc_t66@hus.edu.vn,
dothanhha@ptit.edu.vn, duductien@gmail.com, phantheson@hus.edu.vn

## ABSTRACT

This paper details our submission to the IEEE ICIP 2025 Grand Challenge on the Cityscape Aerial Image Dataset. This dataset includes top-down remote sensing images featuring a diverse range of objects, from large structures like buildings and basketball courts to smaller elements such as vehicles and crosswalks. A key characteristic of this data set is a serious class imbalance. To address this, we propose an object detection framework that leverages advanced data augmentation techniques and state-of-the-art YOLO-based models (specifically, YOLOv11 and YOLOv12). Our core contribution is an augmentation strategy designed to balance class distributions and enhance object diversity. This is achieved by selectively removing, augmenting, and reinserting instances of minority classes, such as tennis courts and crosswalks. We present our model architecture, detail the training convergence, and provide a comprehensive error analysis. This targeted augmentation strategy proved highly effective, resulting in an Average Precision (AP50) of 0.92 for crosswalk, 0.84 for building, and an overall mean Average Precision (mAP50) reach up 0.75 for overall, securing a top-2 ranking in the IEEE ICIP competition.

***Index Terms*****—** Aerial Object Detection, YOLO, Data Augmentation, Cityscape, ICIP 2025.

## 1. INTRODUCTION

Detecting objects in aerial images of urban landscapes is a challenge due to the highly diverse dataset, which includes objects whose sizes range from very small to very large. Moreover, the severe class imbalance makes object localization and recognition difficult. This work, prepared for the ICIP 2025 Challenge, addresses these issues through a robust data-augmentation pipeline and variants of YOLO (YOLOv11 [1], YOLOv12 [2]).

In the field of object detection in aerial images, the DOTA dataset (Xia et al., 2018 [3]) is a large dataset of more than 2,800 high-resolution images (about 4,000×4,000 px), with 188,282 objects, and VisDrone (Mao et al., 2020 [4]) has formulated the key to identify objects and their sizes.

Methods such as RoI Transformer (Ding et al., 2018 [5]) and ReDet (Han et al., 2021 [6]) address orientation variation using an equivariant network, while TPH-YOLO (Zhu et al., 2021 [7]) added a Transformer-based prediction head to handle size variation due to different drone altitudes, along with a convolutional block attention model ( CBAM ) to enhance focus on high object density regions, achieving an AP of 39.18 % on VisDrone2021. However, minimal object detection is still weak due to class imbalance. To overcome this, AMRNet (Wei et al., 2020 [8]) used mask resampling to re-sample objects to balance the number of classes, and PENet (Tang et al., 2020 [9]) used Mask Resampling Module (MRM) to upsample classes with little data, combined two coarse and fine anchor-free detectors (CPEN, FPEN) with Non-maximum Merge (NMM) algorithm to improve small object detection on drone images, the best model achieved an improvement of 8.7% on visDrone.

In terms of data augmentation, general techniques such as Mosaic (Bochkovskiy et al., 2020 [10]), MixUp, CutMix have proven effective on regular images, but in drone images, ColMix (Ly et al., 2023 [11]) uses "image patching" to insert multiple objects into the background without masks, enhancing object intensity better than Mosaic and improving precision-recall; however, it is sensitive to out-of-distribution noise. Messmer et al., 2021 [12] performs automatic image scaling preprocessing, which reduces the influence of measurement variation and speeds up inference by 2–3 times, with successful experimental results on UAVDT, VisDrone, and autonomous data. These methods are still random, which can introduce background noise and reduce realism.

Our main contribution is a two-stage augmentation process: 1) Background generation: remove small objects (e.g., vehicles, crosswalk markings) to create clean backgrounds and rebalance the remaining classes. 2) Object augmentation and reinsertion: apply transformations to the removed objects (rotation, cropping, color jitter) and then paste them back into the prepared backgrounds. We evaluate our models on both the original and the augmented datasets.

Thanks to XYZ agency for funding.

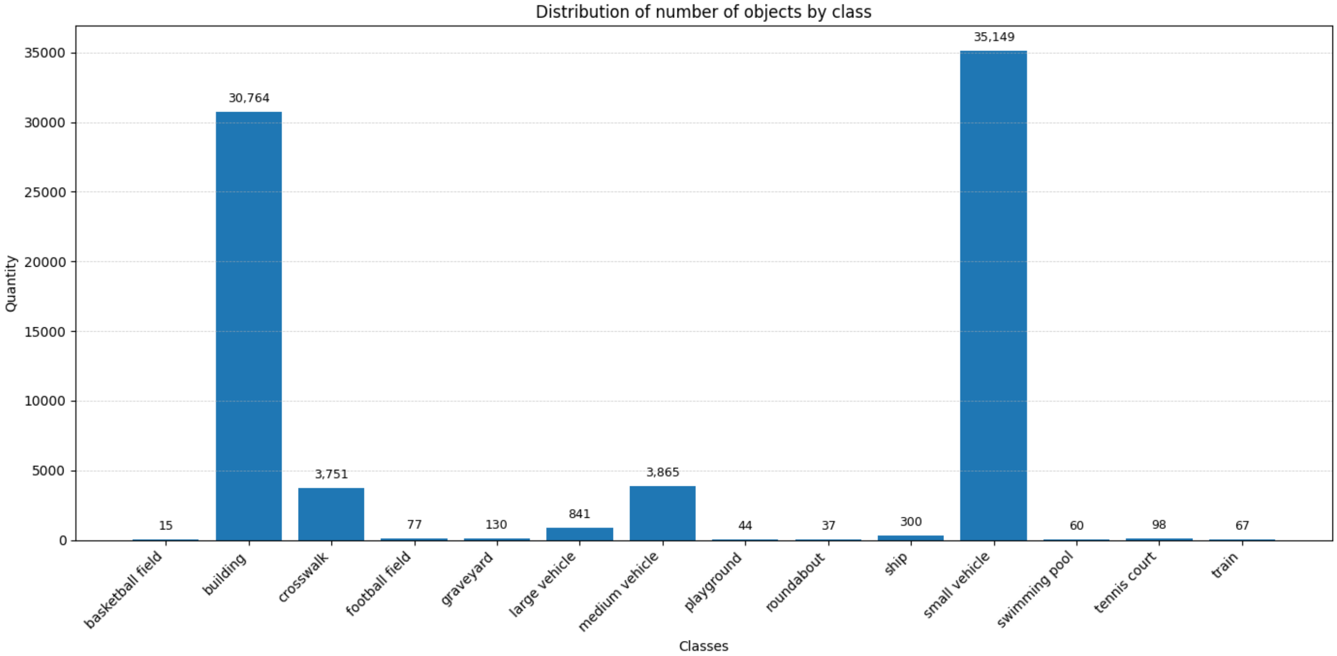


**Fig. 1**: Distribution of classes in the dataset, a noticeable feature of the dataset is the severe imbalance between classes

## 2. DATASET AND AUGMENTATION

The primary data set used in this study is CADOT, which includes aerial cityscape images. This diverse data set is organized in the standard COCO format and includes a total of 3,234 images in the training set, 929 images in the validation set, and 465 images in the test set. Within the training set, there are 75,198 objects distributed in 14 different classes. The data set features objects of various sizes, and there is a significant class imbalance. Figure 1 illustrates the distribution of these classes. Looking at the figure, it is clear that the two classes building and Small vehicle are the two dominant classes with $\approx 40.9\%$ and $\approx 46.7\%$ of the total number of objects, respectively. The classes that can be considered average are medium vehicle ($\approx 5.1\%$), crosswalk ($\approx 5.0\%$) and large vehicle ($\approx 1.1\%$), and the rest account for less than $0.5\%$ each.

Our augmentation aims to improve model robustness and generalization. Additionally, we try to minimize the amount of balancing so that the modeling is not too focused on high-volume objects.

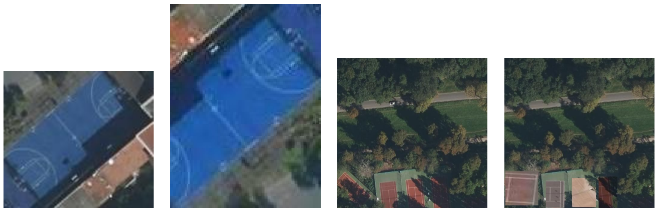
### 2.1. Augmentation Strategy

In this section, we employ a specialized two-phase augmentation strategy. This strategy is crucial for addressing key challenges in aerial imagery, such as significant class imbalance (e.g., numerous vehicles vs. fewer specialized structures) and the need to enhance the diversity of object instances, particularly for smaller or less frequent classes like 'crosswalk', 'ship', or 'train'. Specifically, We present two important strategies for data augmentation:

1. **Background Image Generation:** We first create a set of enhanced background images. This involves selectively removing small, dynamic objects (e.g., 'vehicle', 'crosswalk') from original images by using Inpainting NAVIER_STOKES algorithms [13]. The remaining static objects within these backgrounds are then balanced to ensure a more uniform representation, preventing the model from becoming biased towards overly dominant background features.

2. **Object Augmentation and Re-insertion:** Concurrently, the removed objects, along with additional instances of critical classes, undergo a series of augmentations. These include geometric transformations (rotations, cropping to 2/3 size to simulate scale variation) and photometric transformations (color jittering, saturation adjustments) . These augmented objects are then carefully re-inserted onto the prepared background images or suitable original images, creating rich and diverse training samples.

In Figure 2, the results demonstrate a greater diversity in data regarding size, perspective, and lighting. This enhancement significantly improves the frequency of small classes, aiding in increased recall and balancing accuracy across different classes. However, some feathering and illumination matching still need adjustment to achieve optimal realism.



(a) Original Obj. (b) Augmented Obj. (c) Original BG (d) Augmented BG

**Fig. 2**: Augmentation illustration: (a) Original object, (b) Augmented object, (c) Original background, (d) Processed background.

### 2.2. Generated Datasets for Experiments

Two augmented datasets, Dataset-A (DS-A) and Dataset-B (DS-B), are created from the original dataset (DS-Orig). DS-A is constructed by removing the majority-class objects and repeating the process three times. The resulting images are then combined with two copies of the original dataset, followed by inserting minority-class objects to balance representation. DS-B is generated by replacing easily recognizable objects in DS-Orig with more challenging instances, aiming to improve the model's ability to detect hard-to-recognize classes. The class distributions of both datasets are describe as follows:

**Dataset DS-A (Post Initial Augmentation):** Class counts: basketball field: 22065, football field: 22437, graveyard: 22755, playground: 22239, roundabout: 22197, swimming pool: 22335, tennis court: 22558, ship: 1800, train: 402, large vehicle: 37673, medium vehicle: 46743, crosswalk: 11253, building: 30764, small vehicle: 35149.

**Dataset DS-B (Post Further Augmentation/Refinement):** Class counts: graveyard: 7951, large vehicle: 13399, playground: 7779, medium vehicle: 19446, small vehicle: 46865, swimming pool: 7811, crosswalk: 7502, building: 38455, tennis court: 196, ship: 600, football field: 154, roundabout: 74, train: 134, basketball field: 30.

DS-A exhibits a relatively balanced class distribution. In contrast, DS-B emphasizes rare and hard-to-detect objects, allowing for focused assessment of model performance under challenging data scenarios.

This dual-design approach facilitates a comprehensive evaluation of the model's robustness, particularly in detecting ambiguous or infrequent instances.

The augmentation procedures for both datasets are illustrated in Fig. 3.

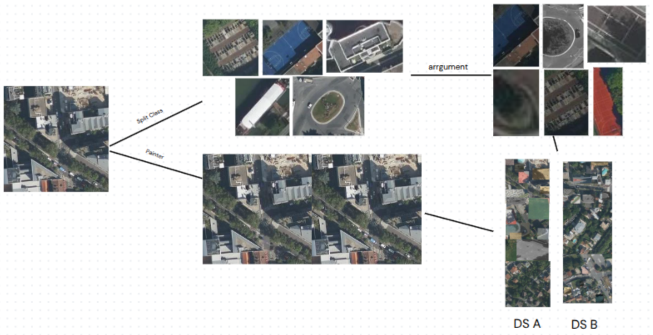


**Fig. 3**: Data augmentation workflow for generating DS-A and DS-B. The original images had been augmentation, including object removal, duplication, and reinsertion, to create DS-A. In DS-B, hard-to-detect objects are inserted to increase detection difficulty and model robustness.

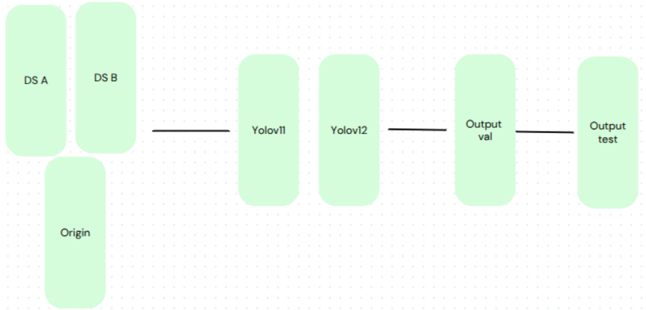


**Fig. 4**: Flow diagram of the object detection pipeline using DS-A, DS-B and DS-Orig. The datasets, derived from the original dataset, are used to train YOLOv1 and YOLOv2 models. Outputs are evaluated on separate validation and test sets.

## 3. METHODOLOGY

Our methodological approach is designed to systematically evaluate the impact of our proposed data augmentation strategies and the performance of different YOLO variants on the aerial cityscape object detection task. The core workflow, depicted in Figure 4, encompasses dataset preparation through augmentation, training of object detection models, and a comprehensive evaluation leading to the selection of the optimal configuration for the final test set.

### 3.1. Object Detection Models

For the object detection task, we selected two advanced variants from the You Only Look Once (YOLO) family: YOLOv11 and YOLOv12. These models were chosen for their state-of-the-art performance, offering an effective trade-off between detection accuracy and computational efficiency—an essential requirement when processing high-resolution aerial imagery.

Their architectures are well-known for capturing multi-scale features, which is particularly beneficial given the wide variation in object sizes within urban environments. We hypothesize that these models, when coupled with our targeted augmentation strategies, can lead to significant improvements in detection performance.

During model training, we configured the hyperparameters as follows: the number of training epochs was set to 100, with a batch size of 64 and an initial learning rate of 1e-2. All input images were resized to a fixed resolution of 640 pixels.

### 3.2. Training and Evaluation Protocol

Each selected YOLO model (YOLOv11 and YOLOv12) was trained independently on the three datasets: the original dataset (DS-Orig), the initially augmented dataset (DS-A), and the further refined augmented dataset (DS-B). Standard training procedures were followed, typically involving initialization with pre-trained weights where applicable, and optimization using appropriate loss functions and learning rate schedules (as shown in Figure 5e).

The performance of each trained model configuration (e.g., YOLOv11 on DS-A, YOLOv12 on DS-B, etc.) was rigorously evaluated on a dedicated validation set, disjoint from the training data. Key metrics, including mean Average Precision (mAP) at various IoU thresholds (e.g., mAP@0.50, mAP@0.50:0.95 as shown in Figure 5d), along with individual class-wise AP scores, were used to compare the effectiveness of different augmentation strategies and model architectures.

Based on this comprehensive validation performance, we identified the model configurations that demonstrated the best overall accuracy and robustness. This involved analyzing convergence behavior (Figures 5a-5c) and the confusion matrix (Figure 6) to understand error patterns. The strategy for the final test set submission, as detailed in Section 3.5, involved selecting the most promising model(s) from this validation phase to predict on the unseen test data. This systematic approach ensures that our final submission is based on empirically validated improvements.

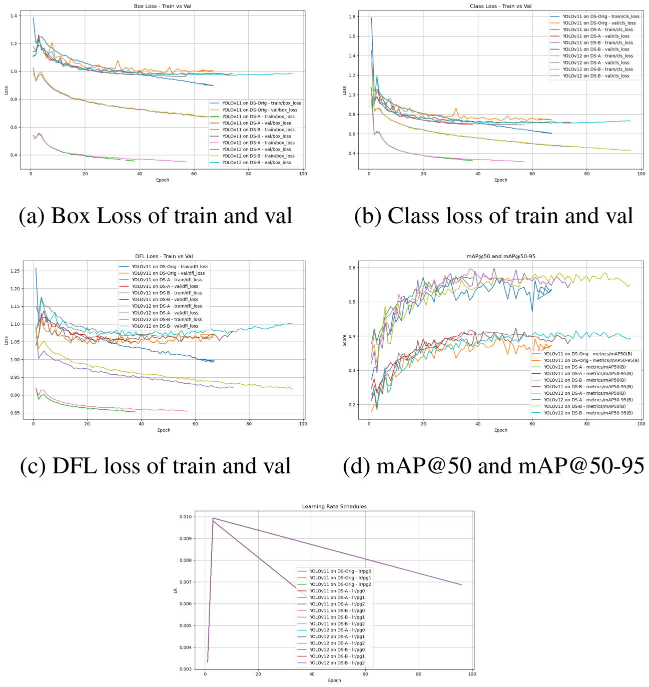
### 3.3. Training Convergence Plots

Convergence is tracked via loss (box, class, DFL) and validation mAP (Figures 5a–5e).

**Convergence Discussion:** The training plots illustrate the convergence behavior of different model configurations (YOLOv11, YOLOv12) across various datasets (DS-Orig, DS-A, DS-B).

Overall, all models demonstrate successful learning, characterized by decreasing loss values (box, class, and DFL; Figures 5a, 5b, 5c) and increasing mAP scores (Figure 5d) over epochs. The learning rate schedules (Figure 5e) were consistent across all experiments, employing a cosine annealing strategy with warm-up, ensuring comparable training dynamics.

A key observation is the performance difference between YOLOv11 and YOLOv12. **Across all three loss metrics, YOLOv12 variants consistently achieved lower final loss values compared to their YOLOv11 counterparts on the same datasets.** This superiority in minimizing loss generally translated to improved detection accuracy.



(a) Box Loss of train and val

(b) Class loss of train and val

(c) DFL loss of train and val

(d) mAP@50 and mAP@50-95

(e) Learning rate schedules

**Fig. 5**: Training Convergence Plots (Loss/mAP vs. Epochs).

The impact of data augmentation (DS-A, DS-B) was model-dependent.

- For **YOLOv11**, the augmented datasets (DS-A, DS-B) did not lead to improved validation mAP scores compared to the original dataset (DS-Orig). In some instances, DS-Orig even showed slightly better or comparable loss and mAP metrics for YOLOv11 (e.g., Figures 5c and 5d).
- Conversely, **YOLOv12 appeared to benefit significantly from data augmentation.** YOLOv12 trained on DS-A consistently yielded the lowest loss values across all types and achieved the highest mAP@50 scores. YOLOv12 on DS-B also outperformed YOLOv12 on DS-Orig in terms of mAP. This suggests that the YOLOv12 architecture is better able to leverage the diversity introduced by augmentation for improved generalization.

Validation losses generally tracked training losses, with small gaps, indicating reasonable generalization. Some validation mAP curves show signs of plateauing towards the later epochs, suggesting that the models were approaching their performance limits under the current training regimen. The consistent learning rate schedules applied (Figure 5e) provide a stable baseline for these comparisons.

**Results and Discussion**

Figure 5d illustrates the progression of mAP@50 and mAP@50-95 over 100 epochs across three dataset configurations (DS-Orig, DS-A, and DS-B) with two model variants (YOLOv11 and YOLOv12). YOLOv12 on DS-B achieves a peak performance of mAP@50 $\approx$ 0.61 and mAP@50-95 $\approx$ 0.45, outperforming all other configurations. Transitioning from DS-Orig to DS-A (background enhancement + class balancing) improves YOLOv11 by approximately +0.15 mAP@50, highlighting the direct impact of balanced data distribution. DS-B, which emphasizes rare class re-synthesis and small object injection, further contributes with noticeable gains: +0.05 mAP@50 for YOLOv11 and +0.04 mAP@50-95 for YOLOv12 compared to DS-A.

Figures 5a–5c present the Distribution Focal Loss (DFL), Class Loss, and Box Loss (train vs. validation), respectively. A clear narrowing of the train–validation gap is observed on DS-A and DS-B, whereas DS-Orig shows signs of overfitting after approximately 50 epochs (training loss continues to drop while validation loss plateaus). DS-A yields the lowest class and box losses, confirming the stabilizing effect of background normalization. Although DS-B exhibits higher absolute losses compared to DS-A, it ultimately achieves superior mAP scores, suggesting that the model tolerates greater localization errors in exchange for enhanced discrimination of hard-to-detect objects.

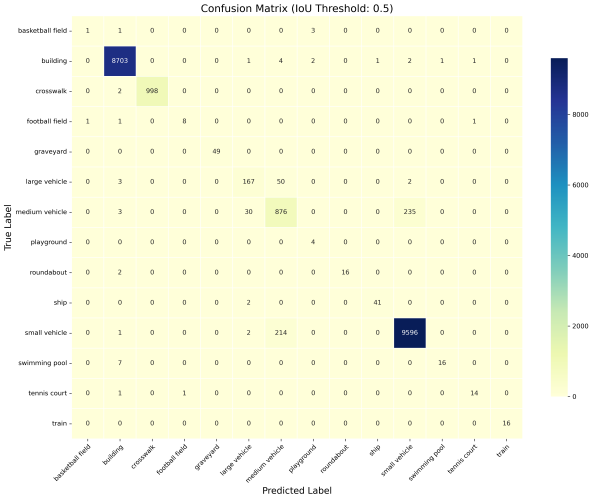
Overall, YOLOv12 + DS-B demonstrates a significant improvement in detection performance (approximately +0.20 mAP compared to the baseline YOLOv11 + DS-Orig) without exacerbating overfitting. These findings underscore the critical role of targeted data augmentation strategies in conjunction with model upgrades and offer a practical pathway for optimizing detection performance on rare objects in urban remote sensing imagery.

### 3.4. Error Analysis

A confusion matrix (Fig. 6) for the best model on the validation set highlights common misclassifications.

**Error Discussion:** A look at the confusion matrix (Figure 6) paints a rather multifaceted picture of the model's performance:

- **Praiseworthy Highlights:** The model demonstrates considerable strength in identifying familiar classes with abundant samples. One can't help but be impressed by the 8703 correct identifications for `building` and a whopping 9596 for `small vehicle`. Even `crosswalk`, with nearly 1000 correct predictions, is a noteworthy achievement. This suggests the model has ”aced the test” for these common objects.



| | basketball field | building | crosswalk | football field | graveyard | large vehicle | medium vehicle | playground | roundabout | ship | small vehicle | swimming pool | tennis court | train |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| basketball field | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| building | 0 | 8703 | 0 | 0 | 0 | 1 | 4 | 2 | 0 | 1 | 2 | 1 | 1 | 0 |
| crosswalk | 0 | 2 | 998 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| football field | 1 | 1 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| graveyard | 0 | 0 | 0 | 0 | 49 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| large vehicle | 0 | 3 | 0 | 0 | 0 | 167 | 50 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| medium vehicle | 0 | 3 | 0 | 0 | 0 | 30 | 876 | 0 | 0 | 0 | 235 | 0 | 0 | 0 |
| playground | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| roundabout | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 0 | 0 | 0 | 0 | 0 |
| ship | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 41 | 0 | 0 | 0 | 0 |
| small vehicle | 0 | 1 | 0 | 0 | 0 | 2 | 214 | 0 | 0 | 0 | 9596 | 0 | 0 | 0 |
| swimming pool | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 0 | 0 |
| tennis court | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 0 |
| train | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 |

**Fig. 6**: Confusion Matrix (IoU Threshold: 0.5) from the model.

- **The ”Field” of Woes:** It's quite disheartening to observe the performance on field-related classes.

  `Basketball field` scored a solitary correct prediction – a figure that's frankly startling. `Football field` (8 correct) and `playground` (4 correct) don't fare much better. It appears distinguishing these types of open areas is a genuinely tough nut for the model to crack.

- **The Perennial Vehicle Size Saga:** Differentiating between `small vehicle`, `medium vehicle`, and `large vehicle` remains a significant, and somewhat predictable, headache. While `small vehicle` is a star performer, the confusion of `medium vehicle` with `small vehicle` (235 instances) and `large vehicle` (30 instances) – and vice-versa – shows the model is still quite muddled when estimating size. The 214 instances where `small vehicle` was mislabeled as `medium vehicle` are also not insignificant.

- **When ’Building’ Becomes a Magnet:** Interestingly, the `building` class seems to exert a peculiar ’gravitational pull’. Several instances of `swimming pool` (7), and even `football field` (1) and `tennis court` (1), were mistakenly ’zoned’ as `building` by the model. Is it possible that when uncertain, the model tends to ’take refuge’ in this highly prevalent class?

- **Commendable Efforts, Yet No Major Breakthroughs:** Classes like `graveyard` (49), `roundabout` (16), `ship` (41), or `train` (16) show the model is certainly trying, but the results remain modest. These are likely ’tough cases’ demanding more finesse and perhaps more distinct features for the model to latch onto.

In summary, this confusion matrix acts as a clear mirror, reflecting both what the model has mastered and where it still stumbles. While there are proud highlights, especially with common objects, the journey to conquer challenging classes and minimize confusion between visually similar ones is evidently still ongoing. This analysis will undoubtedly provide a valuable basis for the development team to strategize and make targeted improvements moving forward.

### 3.5. Test Set Strategy

Building upon the validation phase—where model performance exhibited stability and robustness across critical object classes—we proceeded to identify the top-performing configurations. Selection was informed not only by peak performance metrics but also by consistency and generalization capability, particularly in challenging categories. For the final test stage, we adopted a deployment strategy involving either the single most effective model or an ensemble of models exhibiting complementary behavior, aiming to mitigate class-specific weaknesses while preserving global accuracy.

Upon submission to the official contest platform, the selected model achieved a notable mean Average Precision at IoU threshold 0.50 (mAP@50) of 75.19% on the hidden test set. This result serves as a strong empirical endorsement of the methodological choices in data preprocessing, architecture tuning, and training protocol. It underscores the model's practical efficacy and its potential for real-world deployment in high-resolution remote sensing object detection tasks.

## 4. CONCLUSION

We presented an object detection approach for aerial cityscapes using YOLO variants and a targeted data augmentation pipeline. This strategy, focusing on class balance and instance enrichment, is designed to improve detection accuracy for the ICIP 2025 Challenge.

The strategy yielded very promising results, achieving mAP@50 = 75.19% (on test data set) and securing rank 2 in the ICIP 2025 competition. In the future, we plan to explore integrating generative AI models to further refine and smooth edge cases, enhancing the naturalness of the training data.

# Acknowledgment

We thank the ICIP 2025 organizers for providing the dataset and hosting the challenge.

## 5. REFERENCES

[1] Rahima Khanam and Muhammad Hussain, "Yolov11: An overview of the key architectural enhancements," 2024.

[2] Yunjie Tian, Qixiang Ye, and David Doermann, "Yolov12: Attention-centric real-time object detectors," 2025.

[3] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3974–3983.

[4] Dawei Du, Longyin Wen, Pengfei Zhu, Heng Fan, Qinghua Hu, Haibin Ling, Mubarak Shah, Junwen Pan, Ali Al-Shuwaili, Amr Mohamed, Imene Bakour, Bin Dong, Binyu Zhang, Bouchali Nesma, Chenfeng Xu, Chenzhen Duan, Ciro Castiello, Corrado Mencar, Dingkang Liang, and Zhiyuan Zhao, *VisDrone-CC2020: The Vision Meets Drone Crowd Counting Challenge Results*, pp. 675–691, 08 2020.

[5] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu, "Learning roi transformer for detecting oriented objects in aerial images," *CoRR*, vol. abs/1812.00155, 2018.

[6] Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia, "Redet: A rotation-equivariant detector for aerial object detection," *CoRR*, vol. abs/2103.07733, 2021.

[7] Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," *CoRR*, vol. abs/2108.11539, 2021.

[8] Zhiwei Wei and Chenzhen Duan, "Amrnet: Chips augmentation in areial images object detection," *CoRR*, vol. abs/2009.07168, 2020.

[9] Ziyang Tang, Xiang Liu, Guangyu Shen, and Baijian Yang, "Penet: Object detection using points estimation in aerial images," 01 2020.

[10] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020.

[11] Cuong Ly, Grayson Jorgenson, Dan Rosa de Jesus, Henry Kvinge, Adam Attarian, and Yijing Watkins, "Colmix – a simple data augmentation framework to improve object detector performance and robustness in aerial images," *arXiv preprint arXiv:2305.13509*, 2023.

[12] Martin Messmer, Benjamin Kiefer, and Andreas Zell, "Gaining scale invariance in UAV bird's eye view object detection by adaptive resizing," *CoRR*, vol. abs/2101.12694, 2021.

[13] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, USA, 2000, SIGGRAPH '00, p. 417–424, ACM Press/Addison-Wesley Publishing Co.