International Journal of Multidisciplinary Studies and Innovative Technologies

e-ISSN: 2602-4888
dergipark.org.tr/en/pub/ijmsit
Research Article

2025, 9 (1), 162-172
DOI: 10.36287/ijmsit.9.1.22
Received: July 5, 2025; Accepted: July 27, 2025

Multiple Object Detection and Tracking in Real-Time Aerial Imagery with Deep Learning Architectures

Betül Akyüz[1*], Melih Bahadır[2] and Özkan İnik[3]

[1*] *Tokat Gaziosmanpasa University, Faculty of Engineering and Architecture, Computer Engineering, Tokat, Türkiye (betulakyuz2935@gmail.com) (ORCID: 0009-0005-6106-300X)*
[2] *Tokat Gaziosmanpasa University, Faculty of Engineering and Architecture, Computer Engineering, Tokat, Türkiye (melihbahadir61@gmail.com) (ORCID: 0009-0002-7996-0583)*
[3] *Tokat Gaziosmanpasa University, Faculty of Engineering and Architecture, Computer Engineering, Tokat, Türkiye (ozkan.inik@gop.edu.tr) (ORCID: 0000-0003-4728-8438)*

Abstract – Real-time multi-object detection and tracking is one of the main challenges in analysing aerial imagery from platforms such as unmanned aerial vehicles (UAVs) and satellite systems. Traditional tracking algorithms are inadequate, especially in complex and dynamic environments, which necessitates the development of more powerful and flexible methods. This study proposes a deep learning-based solution aligned with the objectives of high extraction rate and sufficient accuracy. Within the scope of the study, the YOLOv11 model, a single-stage object detection architecture, is integrated with the ByteTrack algorithm for tracking detected objects. This approach is evaluated against the two-stage Faster R-CNN architecture (MobileNetV3-large FPN backbone), known for its high accuracy in object detection. The models were trained on a comprehensive dataset created by combining the challenging VisDrone and DOTA datasets containing objects of various sizes. The results demonstrate that YOLOv11s achieved an mAP@0.5 of 27.7% with an inference speed exceeding 15 FPS, while Faster R-CNN reached a substantially higher mAP@0.5 of approximately 47% but processed frames more slowly, at around 8 FPS. These FPS values represent the number of frames each model can process per second under the same hardware conditions, and do not correspond to the frame rate of the input video. The ByteTrack algorithm, employed in the tracking phase, improves tracking accuracy by successfully identifying detected objects. In this context, the advantages and limitations of both models are evaluated, and it is concluded that the choice of model should be based on the specific application requirements—favoring YOLOv11 for real-time use with limited computational resources, and Faster R-CNN for applications where high detection precision is paramount.

Keywords *– Multi-Object Tracking, YOLOv11, Faster R-CNN, ByteTrack, Aerial Images*

Citation: Akyüz, B., Bahadır, M., & İnik, Ö. (2025). Multiple Object Detection and Tracking in Real-Time Aerial Imagery with Deep Learning Architectures. International Journal of Multidisciplinary Studies and Innovative Technologies, 9(1): 162-172.

I. INTRODUCTION

Today, real-time multi-object detection and tracking (MOT) has gained significant importance in a wide range of applications, including autonomous systems, unmanned aerial vehicles (UAVs), security cameras, and smart city infrastructures. Particularly in dynamic and complex environments—such as aerial and satellite imagery—the ability to accurately, rapidly, and continuously detect and track multiple objects is critical for ensuring the reliability, efficiency, and performance of these systems. However, conventional object detection and tracking algorithms often fall short in scenarios involving rapid object motion, occlusion, or temporary disappearance. Along with the high accuracy rates of deep learning-based models in the field of Classification [1-6] significant success has also been achieved in object detection.

In recent years, novel methods based on the YOLO architecture have emerged as powerful solutions for the efficient and accurate detection of small objects. For instance, the Rotating-YOLO approach improves accuracy while reducing model complexity by addressing rotated objects [7]. PETNet introduces multi-scale feature fusion to minimize information loss in small object detection [8]. EL-YOLO distinguishes itself with a lightweight architecture, enabling fast inference on edge devices [9]. SDMNet leverages attention mechanisms and advanced convolutional techniques to improve detection across varying object sizes [10]. Furthermore, enhancements in YOLOv8 have led to significant improvements in detecting small-scale targets [11]. These studies underscore the strengths of deep learning models in balancing speed and accuracy while tackling key challenges such as small object detection.

In this study, a real-time multi-object detection and tracking system has been developed specifically for use with aerial and satellite imagery. The system evaluates the single-stage, high-speed YOLOv11 architecture against the two-stage Faster R-CNN architecture—employing the MobileNetV3-large FPN

backbone—which is known for its high accuracy in complex scenarios.

The YOLOv11 model offers advantages for real-time applications due to its high inference speed, while the Faster R-CNN model enables more accurate object detection, particularly in complex scenes. The ByteTrack algorithm used for tracking ensures continuity by assigning unique identifiers to objects detected by YOLOv11, enabling reliable tracking of even temporarily invisible or fast-moving objects.

Model training was performed on a challenging and comprehensive dataset created by combining the VisDrone and DOTA datasets, which include objects of different sizes and various environmental conditions. In addition, a lighter YOLOv11 model trained only with the VisDrone dataset was also evaluated. The results of the experiments revealed the performance of each model in terms of accuracy and speed, highlighting their suitability for different application scenarios.

The primary objective of this study is to develop a YOLOv11-based object detection system on datasets containing high-resolution aerial and satellite images, integrate it with the ByteTrack algorithm, and perform multi-object tracking. Additionally, the system's performance will be evaluated by comparing it with the Faster R-CNN architecture, with the goal of comparing the training characteristics, inference speeds, and accuracy performance of different deep learning architectures. In this context, the developed system aims to create a robust infrastructure for the defence industry by enabling the effective use of AI-supported decision-making mechanisms in tasks such as navigation and target tracking for autonomous UAVs.

II. Materials and Method

2.1. Deep Learning Models

In this study, two popular deep learning-based object detection algorithms, YOLOv11 and Faster R-CNN, were used for multi-object detection and tracking from aerial and satellite images. Although these models are based on different artificial intelligence approaches, both provide successful results for object detection in complex scenes.

2.1.1. YOLOv11

YOLOv11 (You Only Look Once version 11) is a recent deep learning architecture introduced by Ultralytics, known for its fast and effective solutions to real-time object detection problems through a single-stage structure. Compared to previous versions, YOLOv11 features a lighter architecture, improved speed, and enhanced small object detection capability. These characteristics make it a balanced model in terms of detection accuracy and inference time, especially in complex aerial and satellite imagery containing multiple objects.

Although YOLOv11 has not yet been presented in a peer-reviewed academic publication, it has been officially released and documented by Ultralytics on their website. This release includes detailed architectural improvements and performance benchmarks, providing a solid basis for research and application in real-time scenarios.

In this study, YOLOv11 is preferred for multi-object detection due to its high performance and updated architecture. Prior research and community applications have also begun to explore its potential. For instance, the LP-YOLO algorithm has been adapted to enhance pedestrian detection efficiency in real-time applications [12], and the SDS-YOLO architecture has shown improvement in tasks requiring positional accuracy under unstable conditions [13]. Additionally, it has been applied successfully in areas such as fire and smoke detection [14].

All these findings, supported by the latest release from Ultralytics, confirm that YOLOv11 is a fast, accurate, and reliable method for multi-object detection in aerial imagery.

2.1.2. Faster R-CNN

Faster R-CNN (Region-based Convolutional Neural Network) is a widely used, region-based two-stage deep learning architecture for object detection tasks. In the first stage of the model, the Region Proposal Network (RPN) is used to identify potential object locations in the image. The RPN generates region proposals (regions with high object probability) using a sliding window approach on convolutional feature maps.

In the second stage, these candidate regions are resized to fixed dimensions using the Region of Interest (RoI) Pooling method, followed by classification and bounding box regression operations. This two-stage structure enables Faster R-CNN to deliver superior performance in applications requiring high accuracy and detailed analysis of small objects.

Although it requires more computational resources compared to single-stage models (YOLO, SSD, etc.), it offers advantages in terms of regional accuracy and classification accuracy. For this reason, Faster R-CNN is preferred in areas such as infrastructure monitoring, agricultural analysis, and high-resolution aerial images, where complex scene structures are present.

For example, it has achieved high accuracy in infrastructure monitoring studies such as detecting bridge cracks in aerial images [15], and has been effectively used in agricultural applications such as identifying weeds [16] and counting wheat heads [17]. In these studies, Faster R-CNN's success in detecting small objects stands out.

2.2. Data Sets

2.2.1.VisDrone

VisDrone is a comprehensive dataset consisting of high-resolution aerial images captured by unmanned aerial vehicles (UAVs) in different urban environments, under various weather conditions and at different density levels. In the field of computer vision, it is considered a critical reference point for tasks such as object detection in aerial images and multi-object tracking (MOT) [18]. In this study, the VisDrone-MOT subset was used, and the dataset was divided into 25,607 images for training, 9,304 for validation, and 6,474 for testing. Fig. 1 presents a typical aerial image from the VisDrone dataset, providing a visual representation of the complex scenes and diverse object categories contained within the dataset.

Fig. 1. An example aerial image from the VisDrone dataset.

VisDrone dataset contains a total of 12 different object classes covering a wide range of objects commonly encountered in urban environments. These classes include 'ignored,' 'pedestrian,' 'people,' 'bicycle,' 'car,' 'van,' 'truck,' 'tricycle,' 'awning-tricycle,' 'bus,' 'motor,' and 'others.' This diversity enables the testing of algorithms' generalisability in real-world scenarios.

The VisDrone dataset is divided into two main sub-sets: VisDrone-DET focuses on object detection tasks on static images, while VisDrone-MOT covers multi-object tracking tasks from sequential video frames. This segmentation provides a unique resource for addressing challenging conditions such as small object detection and object tracking continuity in dynamic scenes, particularly in images captured from a UAV perspective.

The images in the dataset are commonly used to evaluate the performance of small object detection and real-time tracking algorithms (e.g., ByteTrack) due to their low altitude and complex background conditions. This makes VisDrone an important benchmark for testing the practical effectiveness of methods developed in the field of multi-object tracking (MOT).

The challenges presented by VisDrone significantly guide current research and development activities. For example, studies focusing on the domain shift problem in aerial images have provided comprehensive experimental results on the VisDrone dataset, analysing the impact of image- and instance-level shifts on detection performance [18]. Proposed innovations for the YOLOv8 architecture have also been tested on the VisDrone dataset, achieving significant improvements of 10.5% in mAP50 and 6.9% in mAP95, thereby enhancing performance in small target detection [19]. Furthermore, ablation experiments conducted on the VisDrone and AI-TOD datasets improved the metrics of YOLOv8-based anchor-free detectors by 2.2% and 1.9%, respectively, achieving a performance increase of 13.75% and 13.97% compared to the baseline [20]. These developments clearly demonstrate the critical role of VisDrone's challenging structure in enhancing the robustness and accuracy of object detection and tracking algorithms.

2.2.2. DOTA

Object detection in aerial images has gained significant importance in recent years due to strategic applications in civil and military fields [22]. The DOTA dataset plays a fundamental role in developments in this field due to its comprehensive structure consisting of high-resolution aerial images [21]. DOTA successfully reflects complex real-world scenarios by incorporating images obtained from different platforms such as unmanned aerial vehicles (UAVs), satellites, and fixed-wing aircraft [22]. The DOTA dataset is designed to address the unique challenges of object detection in aerial imagery.

In this study, a total of 1,411 images from the DOTA dataset were used for training, 458 for validation, and 937 for testing. The dataset includes classes such as 'plane,' 'ship,' 'storage tank,' 'baseball diamond,' 'tennis court,' 'basketball court,' 'ground track field,' 'harbour,' 'bridge,' 'large vehicle,' 'small vehicle,' 'helicopter,' 'roundabout,' 'soccer ball field,' 'swimming pool,' and 'container crane.' The use of rotated bounding boxes for object labelling has made DOTA a standard reference point for rotated object detection problems. Additionally, its inclusion of small objects at different resolutions provides a critical test environment for research in small object detection [23] [24]

DOTA's contribution to the field lies in its ability to provide a benchmark for the development and evaluation of deep learning-based algorithms. Object detection models in aerial images face challenges such as complex backgrounds, limited long-range contextual information, and high pixel similarity between objects and backgrounds [23] [24]. DOTA provides a standardised platform to objectively evaluate the performance of models aimed at overcoming these challenges. For example, YOLO-based detectors such as YOLO-SM have demonstrated their effectiveness in experiments conducted on the DOTA-v1.0 and DOTA-v1.5 datasets [23]. Additionally, dynamic YOLO-based models incorporating the Luna development mechanism and innovative modules achieved significant improvements in small and flat object detection performance, reaching an average precision (mAP_0.5) of 90.6% on DOTA-v1.5 [24].

The DOTA dataset is considered an indispensable resource for research and development activities in the field of object detection in aerial images. Its rich and challenging data structure provides a critical foundation for testing, comparing, and designing new-generation deep learning-based algorithms. This dataset will continue to play a significant role in shaping future research directions in computer vision problems related to aerial images.

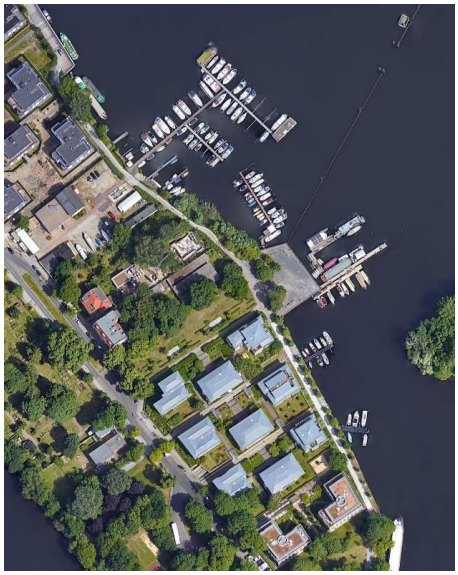

Fig. 2. Example aerial image from the DOTA dataset.

2.3. Performance Metrics

Various metrics are commonly used in computer vision to evaluate the performance of object detection models. These include precision, recall, mean mAP@0.5 (IoU), and mAP@0.5:0.95. Precision refers to the ratio of correctly detected objects to all predictions made by the model, while recall indicates how many of the objects that actually exist were correctly detected [27]. These two metrics allow for the separate evaluation of false positives (false alarms) and false negatives (misses) by the model.

The mAP@0.5 metric measures the average precision of all classes with a specified IoU (Intersection over Union) threshold value of 0.5 and quickly reflects the overall detection performance of the model [26] [28]. However, this fixed threshold value may not fully show how consistent and accurate the model performs at different IoU levels. To address this limitation, the mAP@0.5:0.95 metric is used. This metric calculates IoU values from 0.5 to 0.95 in increments of 0.05, enabling a more comprehensive and detailed analysis of model performance [25] [27] [28]. This allows for a comparison of the model's performance at both loose and strict thresholds, providing a better understanding of its robustness under real-world conditions.

The correct and detailed use of these metrics is critical, especially in challenging tasks such as object detection and tracking in aerial images, due to the density of small objects and class imbalances. Such challenges are among the key factors determining the effectiveness of object detection algorithms in real-world conditions. Metrics such as precision and recall separately highlight false detections and missed objects, enabling the identification of the model's weaknesses, while mAP values summarise the model's overall accuracy and sensitivity.

2.4. Object Tracking Algorithm

ByteTrack is a prominent algorithm in the field of multi-object tracking (MOT) that distinguishes itself by incorporating low-confidence detection results into the tracking process. While traditional MOT methods typically eliminate detections below a certain confidence threshold—leading to the exclusion of small, occluded, or low-resolution objects—ByteTrack takes a different approach. By retaining almost all detection outputs, it improves the association of objects across frames and enables more complete and continuous tracking results.

This strategy aligns with the 'tracking-by-detection' paradigm and significantly enhances the robustness of tracking in dynamic scenes. Unlike conventional approaches, which risk losing critical visual information due to strict filtering, ByteTrack increases reliability by preserving even uncertain detections. This enables the tracking of objects that would otherwise be missed due to partial occlusion or low visual quality[29].

Moreover, ByteTrack can be seamlessly integrated with high-performance object detectors such as YOLOv11 or Faster R-CNN, facilitating precise temporal tracking in real-time systems. Its compatibility with modern deep learning models and its ability to maintain object continuity under challenging conditions make it a highly effective solution for various real-world applications, including autonomous navigation, aerial surveillance, and intelligent monitoring systems.

2.5. Experimental Setup

All training and evaluation processes were conducted using the available hardware infrastructure, which includes NVIDIA GTX 1080 Ti and RTX 3050 GPUs. The YOLOv11 models were trained using the official Ultralytics YOLO implementation, and all models were developed using the Python programming language.

Due to differences in GPU memory capacity—11 GB on the GTX 1080 Ti and 4 GB on the RTX 3050—the datasets were allocated accordingly during the training phase. The relatively smaller VisDrone-DET dataset was trained on the RTX 3050 GPU, while the training involving the combined VisDrone-MOT and DOTA datasets was carried out on the GTX 1080 Ti to meet the higher memory requirements.

III.EXPERIMENTAL STUDIES

In this study, YOLOv11 and Faster R-CNN models were employed for multi-object detection. The models were trained and evaluated on the VisDrone and DOTA datasets, and a lightweight YOLOv11 variant trained solely on the VisDrone-DET dataset was also included for comparison.

Following the detection stage, the outputs of the YOLOv11 model were processed using the ByteTrack algorithm to assign unique IDs to objects, enabling consistent tracking across frames. All tracking results were visually evaluated, and the system’s real-time performance was tested.

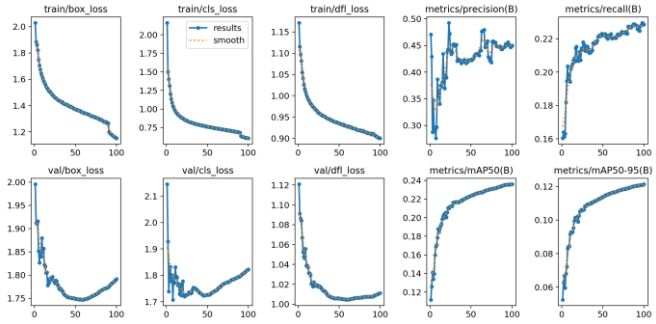

Fig. 3. YOLOv11n Training Convergence Graphs

Fig. 3. presents the performance metrics of the YOLOv11n model during training with a batch size of 4 and an image size of 640x640 pixels for 100 epochs. While the training losses show a steady decline, the validation losses (especially after the 60th epoch) exhibit a slight upward trend.

This indicates potential overfitting of the model to the training data. When examining the final performance metrics of the model, precision is acceptable at 0.44964, while recall is quite low at 0.2285. The average precision values, mAP50(B) 0.23604 and mAP50-95(B) 0.12141, indicate that the model's overall detection capability and, in particular, the accuracy of bounding box positioning need to be improved. The low sensitivity and mAP50-95 values reveal that the model misses many real objects and is insufficient in precise positioning.

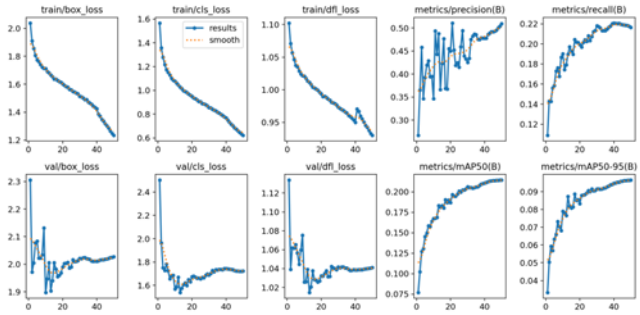

Fig. 4. YOLOv11s Training Convergence Graphs

Fig. 4. presents the performance metrics of the YOLOv11s model during training with a batch size of 4 and an image size of 640x640 pixels over 100 epochs. While training losses show a steady decline, validation losses (val/box_loss, val/cls_loss, val/dfl_loss) have stabilised relatively after the initial drop; this indicates a less pronounced overfitting tendency compared to the previous YOLOv11n model.

When examining the final performance metrics of the model at 100 epochs, precision was recorded as 0.48174. Sensitivity (recall) was 0.25666, mAP50(B) was 0.27768, and mAP50-95(B) was 0.15492, showing significant improvements in all metrics compared to the previous YOLOv11n model. In particular, the increase in recall and mAP50-95 values indicates that the model can detect more objects and achieve a significant improvement in the positional accuracy of bounding box predictions. These findings confirm that YOLOv11s offers a more effective solution in terms of object detection performance.

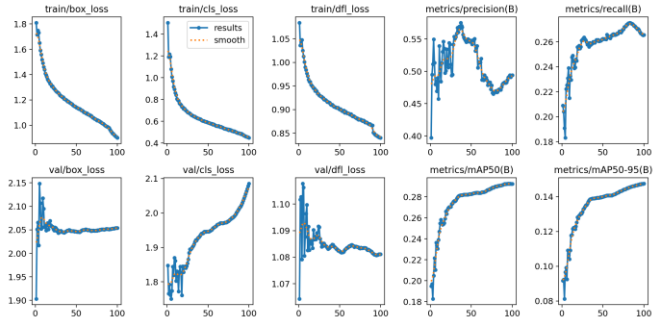

Fig. 5. YOLOv11m Training Convergence Graphs

Fig. 5. summarises the training process of the YOLOv11m model over 100 epochs, with a batch size of 1 and an image size of 640x640 pixels. While the training losses decrease, a significant increase in the validation classification loss (val/cls_loss) is observed after approximately 30 epochs, indicating overfitting.

The final metric results of the model are precision 0.4942, recall 0.26545, mAP50(B) 0.29258, and mAP50-95(B) 0.14758. These values show an improvement over the previous YOLOv11s model, particularly in terms of recall and mAP50(B), but indicate that position accuracy still needs to be improved at high IoU thresholds (mAP50-95). The single batch size (batch size 1) may have a limiting effect on the model's optimisation stability and generalisation performance. Strategies such as larger batch sizes and early stopping are recommended to reduce overfitting and improve performance.

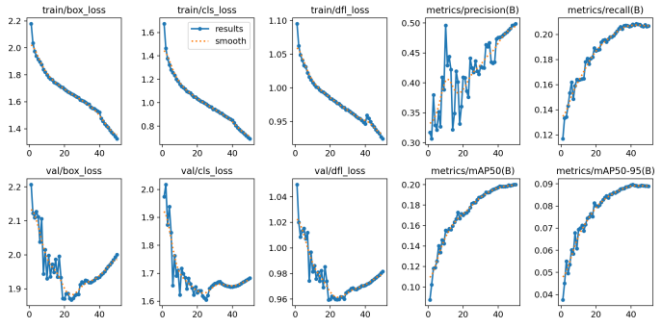

Fig. 6. YOLOv11l Training Convergence Graphs

Fig. 6. shows the training performance of the YOLOv11l model with 50 epochs, a batch size of 4, and an image size of 416x416 pixels. While training losses show a decreasing trend, the slight increase observed in validation losses (val/box_loss, val/cls_loss) after approximately 25 epochs indicates a potential overfitting issue.

The final metric results of the model are precision 0.49895, recall 0.20676, mAP50(B) 0.20003, and mAP50-95(B) 0.08892. These values indicate that the model's accuracy is moderate, but sensitivity and, in particular, positioning accuracy at high IoU thresholds (mAP50-95) need to be significantly improved. The model's overall performance is lower compared to larger YOLO models, particularly in terms of recall and mAP values. Strategies such as early stopping and hyperparameter optimisation are recommended to reduce overfitting and improve model performance

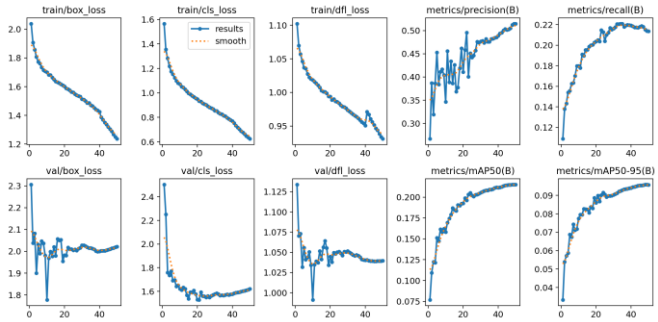

Fig. 7. YOLOv11l Training Convergence Graphs

Fig. 7. summarises the training performance of the YOLOv11l model with 50 epochs, a batch size of 8, and an image size of 480x480 pixels. While the training losses show the expected decrease, the slight increase observed in the validation classification loss (val/cls_loss) after approximately 25 epochs indicates a potential overfitting issue.

The model's final metric results are relatively good with precision at 0.51493, but recall at 0.21355 and mAP50(B) at 0.21512 are moderate, while mAP50-95(B) at 0.09579 is low. These values indicate that the model's object detection accuracy is acceptable, but it requires significant improvement in terms of sensitivity and, especially, localization accuracy at high IoU thresholds. Strategies such as early stopping and hyperparameter optimisation are recommended to reduce overfitting and improve model performance.

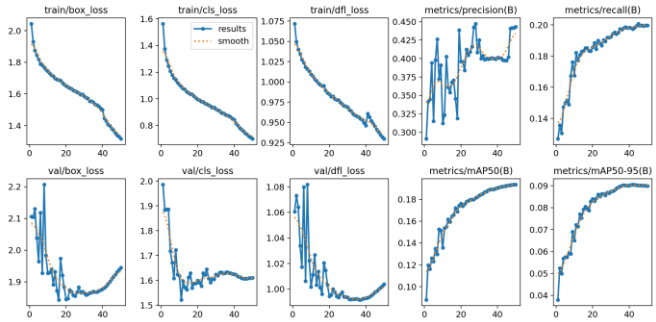

Fig. 8. YOLOv11s Training Convergence Graphs

Fig. 8. presents the training results obtained by the YOLOv11s model over 50 epochs with a 480x480 image size, a batch size of 10, and a final learning rate of 0.001. While training losses show a decreasing trend, the slight increase observed in the validation classification loss (val/cls_loss) after approximately 25 epochs indicates a potential overfitting issue. The model's final metrics (precision: 0.44296, recall: 0.19971, mAP50(B): 0.1935, mAP50-95(B): 0.08987) demonstrate lower performance compared to previous YOLOv11 configurations. In particular, the low precision and mAP values indicate that significant improvements are needed in the model's detection coverage and precise localisation ability. The combination of hyperparameters used (especially the large batch size and relatively short training time) may have contributed to this decline. Strategies such as early stopping and hyperparameter optimisation are recommended to improve performance.

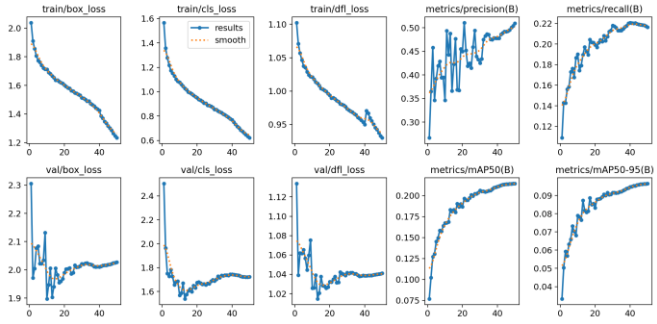

Fig. 9. YOLOv11l Training Convergence Graphs

The graphs shown in Fig. 9. and the obtained metric results reflect the training performance of the YOLOv11l model with a 480x480 pixel image size, a batch size of 8, and a final learning rate of 0.001 over 50 epochs. While the training losses show the expected decrease, the slight increase observed in the validation classification loss (val/cls_loss) after approximately 25 epochs indicates a potential overfitting issue. The model's final metrics are recorded as precision 0.50947, recall 0.21647, mAP50(B) 0.21431, and mAP50-95(B) 0.09652. These values indicate that the model's detection sensitivity is relatively acceptable, but there is a need for significant improvements in sensitivity and, in particular, object localisation accuracy (mAP50-95) at high IoU thresholds. Considering the observed overfitting tendency and relatively low performance metrics, it is recommended to apply strategies such as early stopping and more comprehensive hyperparameter optimisation to enhance the model's generalisation ability.

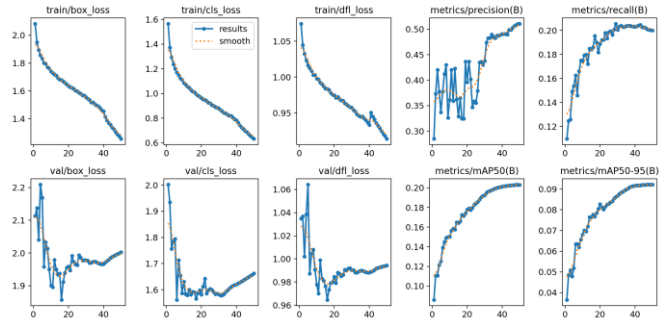

Fig. 10. YOLOv11l Training Convergence Graphs

The graphs and metric results presented in Fig. 10. summarise the training performance of the YOLOv11l model over 50 epochs with a 416x416 pixel image size, a batch size of 10, and a final learning rate of 0.001. While the training losses show the expected decrease, the slight increases observed in the validation losses (val/box_loss, val/cls_loss) after approximately 25 epochs indicate a potential overfitting issue. The model's final metrics are recorded as precision 0.51083, recall 0.20005, mAP50(B) 0.20298, and mAP50-95(B) 0.09214. These values indicate that the model's detection accuracy is relatively acceptable, but there is a need for significant improvements in sensitivity and, in particular, object localisation accuracy (mAP50-95) at high IoU thresholds. Considering the observed overfitting tendency and relatively low performance metrics, especially when using a large batch size of 10 and a relatively small image size (416x416), it is recommended to apply strategies such as early stopping and hyperparameter optimisation to enhance the model's generalisation ability.

Model	Epoch	Batch - Size	İmage - Size	lr0	lrf	mAP50(B)	mAP50-95(B)	precision(B)	recall(B)
YOLOv11n	100	4	640	0.01	0.01	0.23604	0.12141	0.44964	0.2285
YOLOv11s	100	4	640	0.01	0.01	0.27768	0.15492	0.48174	0.25666
YOLOv11m	100	1	640	0.01	0.01	0.29258	0.14758	0.4942	0.26545
YOLOv11l	50	4	416	0.01	0.01	0.20003	0.08892	0.49895	0.20676
YOLOv11l	50	8	480	0.01	0.01	0.21512	0.09579	0.51493	0.21355
YOLOv11s	50	10	480	0.01	0.001	0.1935	0.08987	0.44296	0.19971
YOLOv11l	50	8	480	0.01	0.001	0.21431	0.09652	0.50947	0.21647
YOLOv11l	50	10	416	0.01	0.001	0.20298	0.09214	0.51083	0.20005

Table 1. Training Parameters with the YOLOv11 Model

As detailed in Table 1, various YOLOv11 model variants were trained under different configurations, including varying batch sizes, image resolutions, and number of epochs. To improve the models' generalisation ability and robustness against variations in input data, a comprehensive set of data augmentation techniques was employed consistently across all training experiments.

Specifically, horizontal flipping was applied with a probability of 0.5 (fliplr=0.5), while vertical flipping was disabled (flipud=0.0). In addition, the training pipeline included random rotations up to ±10 degrees (degrees=10), as well as color space augmentations in the HSV domain—

namely, hue adjustment within ±0.015 (hsv_h=0.015), saturation variation up to ±0.7 (hsv_s=0.7), and brightness variation up to ±0.4 (hsv_v=0.4). Geometric transformations such as random scaling (scale=0.8, i.e., between 0.8 and 1.2) and translation (translate=0.1, i.e., up to ±10% of the image dimensions) were also incorporated. These augmentation parameters were activated using the augment=True setting.

The consistent application of these augmentation strategies aimed to increase the diversity of training samples artificially, thereby enhancing the models' performance under real-world conditions. The resulting detection metrics—such as precision, recall, mAP50, and mAP50-95—demonstrate the impact of these techniques, as shown in Table 1.

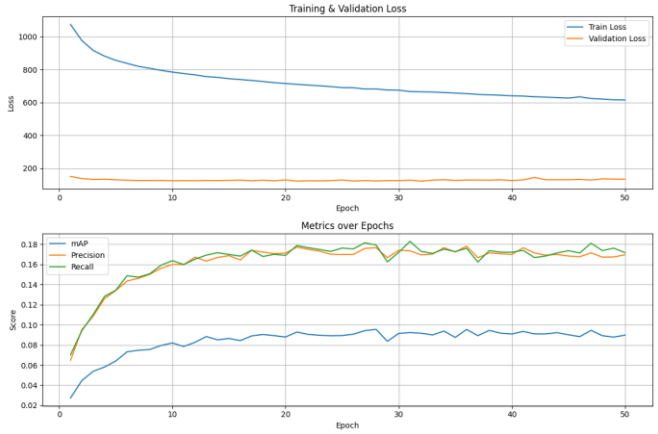

Fig. 11. Faster R-CNN Training Convergence Graphs

The Faster R-CNN model has demonstrated remarkable performance in object detection tasks. Following a training process of 50 epochs, the model achieved average precision (mAP) values of mAP@0.5≈47% and mAP@0.5:0.95≈20.2%. However, the model's inference speed is limited to an average of 8 frames per second (FPS), which poses a constraint for real-time system integration. The training graph is shown in Fig. 11.

When examining the training dynamics, a steady decrease in the loss functions (both training and validation losses) was observed over 50 epochs. This trend indicates that the model successfully converged to the data distribution and maintained its generalisation ability without a significant increase in overfitting. The precision and recall metrics showed a relatively slow increase in the early stages of the training process, but followed a noticeable upward trend from approximately the 10th epoch onwards, indicating a gradual improvement in the model's target detection capability. The average precision (mAP) value also exhibited a similar trend.

The model was trained on images with a resolution of 512x512 pixels, processed in batches of 8 examples each. The optimisation algorithm used was Stochastic Gradient Descent (SGD), with an initial learning rate of 0.005 and a final learning rate of 0.01.

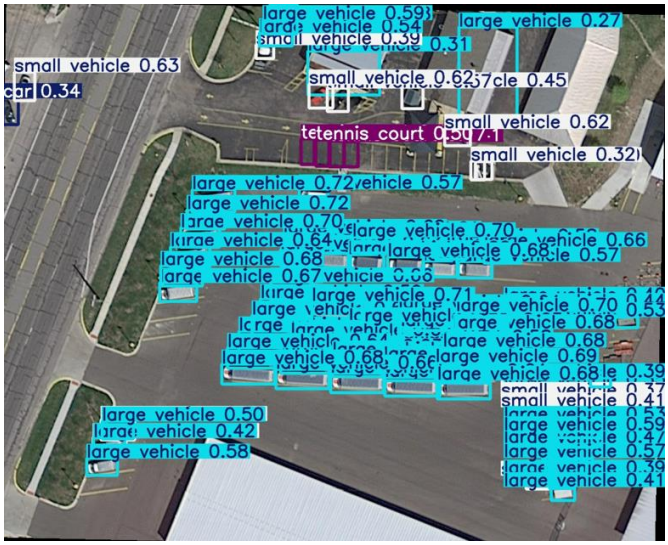

Fig. 12. Object Detection with YOLOv11

An example image of object detection performed with the YOLOv11 model is shown in Fig. 12.

Fig. 13. Faster R-CNN Object Detection

Fig. 13 shows an image of object detection performed with Faster R-CNN.

Fig. 14. Object Tracking with ByteTrack

Fig. 14 shows a visual representation of tracking performed with ByteTrack.

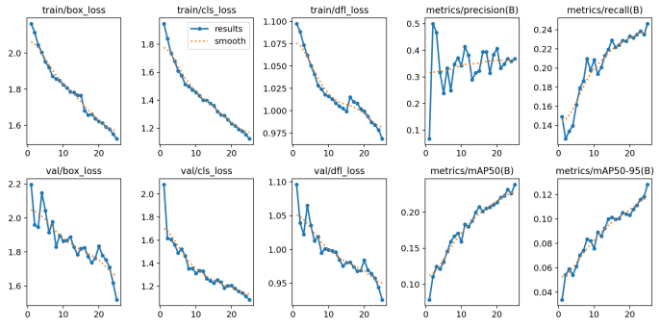

Fig. 15. YOLOv11s Training Convergence on VisDrone-DET Dataset

The graphs and metric results presented in Fig. 15. detail the training performance of the YOLOv11s model on the VisDrone DET dataset, using a 480x480 pixel image size, a batch size of 10, and a final learning rate of 0.001 over 25 epochs. The training losses (train/box_loss, train/cls_loss, train/dfl_loss) exhibit the expected decreasing trend throughout the training period, while the validation losses (val/box_loss, val/cls_loss, val/dfl_loss) show no significant overfitting trend during this short training period. The final performance metrics of the model are recorded as precision 0.36863, recall 0.24646, mAP50(B) 0.23819, and mAP50-95(B) 0.12821. These values indicate that the model's overall detection capability is moderate, but significant improvements are needed in both object coverage (recall) and, especially, bounding box positioning accuracy at high IoU thresholds (mAP50-95). It is considered that the performance achieved can be attributed to the limited training time (25 epochs) and that the model has potential for further optimisation.

In this study, three different training scenarios were considered for object detection and tracking tasks. First, YOLOv11 models were trained on a combination of the VisDrone MOT and DOTA datasets. Second, the same combined dataset was trained using the Faster R-CNN model. Third, the YOLOv11 model was trained only on the VisDrone DET dataset. The training outputs of all three models were carefully evaluated graphically in terms of metrics such as Precision, Recall, mAP, and loss, and recorded throughout the training. Finally, object tracking was performed using the ByteTrack algorithm based on the results obtained from these detection models, and the tracking results demonstrated that objects were reliably identified and high accuracy was achieved during the tracking process.

IV. DISCUSSION

This study offers a comprehensive assessment of contemporary deep learning-based object detection and tracking paradigms, with a particular emphasis on real-time applications. Within this framework, both single-stage (YOLOv11 variants) and two-stage (Faster R-CNN) detection architectures were evaluated using a combined dataset comprising VisDrone and DOTA. Additionally, standalone experiments on the VisDrone DET dataset were conducted exclusively with YOLOv11 models to investigate the performance of lightweight architectures in resource-constrained scenarios.

Following the detection phase, the ByteTrack algorithm was utilized to perform multi-object tracking. Its integration enabled reliable and continuous tracking by maintaining consistent object identities, even in the presence of rapid motion or temporary occlusion.

The training dynamics of different-sized (n, s, m, l) and configured models of the YOLOv11 family were examined in detail through the evolution of loss functions. In general, training losses showed a steady decrease across all variants, indicating that the models effectively learned the patterns in the training data. However, changes in validation losses provided critical insights into the models' generalisation ability. For example, in the 100-epoch training of the YOLOv11n model, a slight increase in validation losses starting after approximately 60 epochs indicated a potential overfitting tendency. This situation carried the risk of negatively affecting the model's performance on test data by over-adapting to the training data. The final metrics, particularly the low sensitivity (0.20005) and mAP50-95(B) (0.09214) values, indicated significant limitations in the model's target detection coverage and bounding box positioning accuracy.

In the 100-epoch training of the YOLOv11s model, validation losses followed an initial decline and then stabilised relatively, indicating better generalisation ability. This model showed significant improvements over the previous YOLOv11n in all basic metrics (precision: 0.48174, recall: 0.25666, mAP50(B): 0.27768, mAP50-95(B): 0.15492), proving that more objects can be detected and that there is a significant increase in location accuracy.

When trained with a small batch size of 1, the YOLOv11m model showed serious overfitting symptoms with a significant increase in validation classification loss (val/cls_loss) after approximately 30 epochs. This indicates that small batch sizes may negatively affect training stability and limit generalisation ability. Metrics (precision: 0.4942, recall: 0.26545, mAP50(B): 0.29258, mAP50-95(B): 0.14758) showed an improvement in mAP50 compared to the previous YOLOv11s model, but it was emphasised that accuracy and positional accuracy at high IoU thresholds still need to be improved.

Different 50-epoch configurations of the YOLOv11l model were also investigated. Training with a 416x416 image size and a batch size of 4 (precision: 0.49895, recall: 0.20676, mAP50(B): 0.20003, mAP50-95(B): 0.08892), slight overfitting symptoms were observed after approximately 25 epochs in validation losses. When a larger image size (480x480) and a batch size of 8 were used (precision: 0.51493, recall: 0.21355, mAP50(B): 0.21512, mAP50-95(B): 0.09579), while precision improved slightly, sensitivity and mAP50-95 values still require further improvement. Reducing the final learning rate (lrf) to 0.001 (for the 480x480, batch 8 configuration) did not result in significant changes in the metrics, and the overfitting tendency persisted (precision: 0.50947, recall: 0.21647, mAP50(B): 0.21431, mAP50-95(B): 0.09652). Similarly, the performance of the YOLOv11l model trained with a 416x416 image size and a batch size of 10 (lrf=0.001) also showed signs of overfitting with low precision and mAP values (precision: 0.51083, recall: 0.20005, mAP50(B): 0.20298, mAP50-95(B): 0.09214) showed signs of overfitting with low precision and mAP values. The YOLOv11s model trained on the VisDrone DET dataset for 25 epochs did not show significant overfitting in its losses within this short timeframe; however, its metric values (precision:

0.36863, recall: 0.24646, mAP50(B): 0.23819, mAP50-95(B): 0.12821) indicate that the model still has significant optimisation potential.

The Faster R-CNN model has demonstrated very high performance in object detection tasks. Following a 50-epoch training process, it achieved average accuracy (mAP) values of mAP@0.5≈47% and mAP@0.5:0.95≈20.2%. During training, it was observed that the loss functions continuously decreased and the model maintained its generalisation ability without a significant increase in overfitting. The accuracy and recall metrics also showed a noticeable upward trend starting from approximately the 10th epoch. However, the inference speed of Faster R-CNN is limited to an average of 8 frames per second (FPS), which poses a significant constraint for real-time system integration. The training of this model was performed on 512x512 pixel image size and 8-sample mini-batches.

In the object tracking phase of the study, the ByteTrack algorithm processed the detection outputs from the YOLOv11 model to assign identities to objects and enable tracking across different frames. The tracking results demonstrated that objects were reliably identified and high accuracy was achieved during the tracking process. This finding confirms that ByteTrack, integrated with a powerful detection model, can provide reliable multi-object tracking even in dynamic and complex scenes.

In general, the YOLOv11 and Faster R-CNN models offer unique advantages and disadvantages in object detection tasks. Faster R-CNN excels in detection accuracy with higher mAP values but is limited in real-time applications due to its lower inference speed. YOLOv11 family models, while generally achieving higher inference speeds compared to Faster R-CNN, exhibit performance variability depending on different sizes and training configurations. Overfitting issues and lower sensitivity/mAP values are particularly noticeable in variants with small models (YOLOv11n) or specific hyperparameter combinations (e.g., single batch size in YOLOv11m). Larger YOLOv11 variants (s, l) demonstrate better overall performance while maintaining improvement potential, particularly in fine-grained localization (mAP50-95) metrics. These results highlight the importance of carefully balancing accuracy and speed depending on application requirements. The overfitting tendencies observed during training highlight the importance of early stopping, more advanced learning rate schedules, and enriched data augmentation techniques. This comprehensive evaluation on diverse datasets of varying scales and complexities, such as VisDrone and DOTA, has provided valuable insights into the models' adaptability to different scenarios. Finally, the successful integration of outputs from detection models with the ByteTrack algorithm demonstrates the practical applicability of multi-object tracking in fields such as autonomous systems and surveillance applications. Future work could focus on architectural optimisations and custom loss functions to improve the performance of YOLOv11 models at high sensitivity and high IoU thresholds. Additionally, minimising accuracy loss while increasing inference speeds through model compression and quantisation techniques will open up broader application areas in real-time systems.

V. CONCLUSION

This study aims to comprehensively examine the performance of current deep learning-based approaches, such as the YOLOv11 family of models and Faster R-CNN, in the field of multi-object detection and tracking using the VisDrone and DOTA datasets. The findings provide valuable insights into the capabilities and limitations of these models, while also evaluating the impact of integrating detection algorithms with ByteTrack on tracking performance.

Experimental results reveal that different variants and hyperparameter configurations of YOLOv11 models exhibit variable performance. While training losses generally show a consistent decrease, significant overfitting tendencies are observed in validation losses, particularly during long training periods or with small batch sizes (e.g., a batch size of 1 in YOLOv11m). This indicates that the models are overly adapted to the training data, limiting their generalisation ability. In particular, low sensitivity and mAP50-95 values in YOLOv11n and some YOLOv11l configurations indicate that the models are unable to detect all objects and localise them with high accuracy. In contrast, the 100-epoch training of the YOLOv11s model demonstrated more stable validation losses and significant improvements in overall metrics (precision: 0.48174, recall: 0.25666, mAP50(B): 0.27768, mAP50-95(B): 0.15492), demonstrating more effective performance. The Faster R-CNN model achieved very high mAP values (mAP@0.5≈47%, mAP@0.5:0.95≈20.2%) in the object detection task, demonstrating strong performance in terms of accuracy.

The continuous decrease in loss functions and gradual improvement in metrics observed during training support the model's strong generalisation ability. However, the inference speed of 8 frames per second (FPS) on average poses a significant constraint for real-time system integration.

The ByteTrack algorithm, an integral part of the study, demonstrated successful performance in object tracking using outputs from detection models. The tracking results showed that objects were reliably identified and high accuracy was achieved during the tracking process. This finding confirms that ByteTrack, combined with a robust detection model, offers an effective solution for reliable multi-object tracking in dynamic environments.

In conclusion, this study has thoroughly demonstrated the object detection capabilities of both the YOLOv11 family and Faster R-CNN models, highlighting the fundamental trade-offs between speed and accuracy. The excessive convergence tendencies and metric variability in the training processes of different YOLOv11 variants confirm the critical role of strategies such as hyperparameter optimisation and early stopping. While Faster R-CNN achieves superior detection accuracy, its lower inference speed limits its applicability in certain domains.

The successful integration with ByteTrack's detection outputs demonstrates the feasibility of building a comprehensive object detection and tracking system. Future work may explore more advanced data augmentation techniques, learning rate schedules, and architectural improvements to improve the performance of YOLOv11 models, particularly on critical metrics such as sensitivity and mAP50-95.

Additionally, improving inference speeds through model compression and hardware acceleration methods for high-

accuracy models like Faster R-CNN could expand their potential in real-time applications. Specialised detection heads or multi-scale feature fusion approaches for detecting small and dense objects in complex datasets such as VisDrone could also be a focus of future research. Finally, advanced metrics and comparative analyses could be used to more comprehensively evaluate the performance of tracking algorithms under different challenges (e.g., long-term occlusions, identity change scenarios).

ACKNOWLEDGMENT

This study was conducted as part of the undergraduate capstone project in the Department of Computer Engineering at Tokat Gaziosmanpaşa University. It was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under the 2209-A Research Projects Support Programme for Undergraduate Students (Project No: 1919B012406548). The authors express their sincere gratitude to all individuals and institutions who contributed to the successful completion of this research.

Authors' Contributions

The authors' contributions to the paper are equal.

Statement of Conflicts of Interest

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The authors declare that this study complies with Research and Publication Ethics

REFERENCES

[1] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* 1409.1556.].

[2] Inik, O., İnik, Ö., Öztaş, T., Demir, Y., & Yüksel, A. (2023). Prediction of soil organic matter with deep learning. Arabian Journal for Science and Engineering, 48(8), 10227-10247.

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

[4] İnik, Ö., Uyar, K., & Ülker, E. (2018). Gender classification with a novel convolutional neural network (CNN) model and comparison with other machine learning and deep learning CNN models. Journal Of Industrial Engineering Research, 4(4), 57-63.

[5] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[6] İnik, Ö., & Turan, B. (2018). Classification of animals with different deep learning models. Journal of New Results in Science, 7(1), 9-16.

[7] Liu, Z., Chen, Y., & Gao, Y. (2024). Rotating-YOLO: A novel YOLO model for remote sensing rotating object detection. Image and Vision Computing, 105397.

[8] Wang, T., Ma, Z., Yang, T., & Zou, S. (2023). PETNet: A YOLO-based prior enhanced transformer network for aerial image detection. Neurocomputing, 547, 126384.

[9] Xue, C., Xia, Y., Wu, M., Chen, Z., Cheng, F., & Yun, L. (2024). EL-YOLO: An efficient and lightweight low-altitude aerial objects detector for onboard applications. Expert Systems with Applications, 256, 124848.

[10] Battish, N., Kaur, D., Chugh, M., & Poddar, S. (2024). SDMNet: spatially dilated multi-scale network for object detection for drone aerial imagery. Image and Vision Computing, 150, 105232.

[11] Luo, W., & Yuan, S. (2025). Enhanced YOLOv8 for small-object detection in multiscale UAV imagery: Innovations in detection accuracy and efficiency. Digital Signal Processing, 158, 104964.

[12] Qu, Z., Liu, H., Kong, W., Gu, J., Wang, C., Deng, L., ... & Lin, F. (2025). LP-YOLO: An improved lightweight pedestrian detection algorithm based on YOLOv11. *Digital Signal Processing*, 105343.

[13] Wang, D., Tan, J., Wang, H., Kong, L., Zhang, C., Pan, D., ... & Liu, J. (2025). SDS-YOLO: An improved vibratory position detection algorithm based on YOLOv11. *Measurement*, *244*, 116518.

[14] Ramos, A., Moraes, F., & Martins, R. (2023). Fire and smoke detection in ground and aerial images using YOLO-based architectures. *Fire Technology*, 59, 2091–2113.

[15] Li, R., Yu, J., Li, F., Yang, R., Wang, Y., & Peng, Z. (2023). Automatic bridge crack detection using Unmanned aerial vehicle and Faster R-CNN. Construction and Building Materials, 362, 129659.

[16] Cui, J., Zhang, X., Zhang, J., Han, Y., Ai, H., Dong, C., & Liu, H. (2024). Weed identification in soybean seedling stage based on UAV images and Faster R-CNN. *Computers and Electronics in Agriculture*, *227*, 109533.

[17] Li, L., Hassan, M. A., Yang, S., Jing, F., Yang, M., Rasheed, A., ... & Xiao, Y. (2022). Development of image-based wheat spike counter through a Faster R-CNN algorithm and application for genetic studies. The Crop Journal, 10(5), 1303-1311.

[18] Ma, Y., Chai, L., Jin, L., & Yan, J. (2024). Hierarchical alignment network for domain adaptive object detection in aerial images. ISPRS Journal of Photogrammetry and Remote Sensing, 208, 39-52.

[19] Luo, W., & Yuan, S. (2025). Enhanced YOLOv8 for small-object detection in multiscale UAV imagery: Innovations in detection accuracy and efficiency. Digital Signal Processing, 158, 104964.

[20] Gu, Q., Han, Z., Kong, S., Huang, H., Li, Y., Fan, Q., & Wu, R. (2025). DCYOLO: Dual negative weighting label assignment and cross-layer decouple head for YOLO in remote sensing images. Expert Systems with Applications, 281, 127595.

[21] Chen, Z., Wang, H., Wu, X., Wang, J., Lin, X., Wang, C., ... & Li, D. (2024). Object detection in aerial images using DOTA dataset: A survey. International Journal of Applied Earth Observation and Geoinformation, 134, 104208.

[22] Nguyen, K., Huynh, N. T., Le, D. T., Huynh, D. T., Bui, T. T. T., Dinh, T., ... & Nguyen, T. V. (2025). A comprehensive review of few-shot object detection on aerial imagery. Computer Science Review, 57, 100760.

[23] Han, L., Li, N., Zhong, Z., Niu, D., & Gao, B. (2025). Adaptive scale matching for remote sensing object detection based on aerial images. Image and Vision Computing, 157, 105482.

[24] Jobaer, S., Tang, X. S., & Zhang, Y. (2025). A deep neural network for small object detection in complex environments with unmanned aerial vehicle imagery. Engineering Applications of Artificial Intelligence, 148, 110466.

[25] Xue, C., Xia, Y., Wu, M., Chen, Z., Cheng, F., & Yun, L. (2024). EL-YOLO: An efficient and lightweight low-altitude aerial objects detector for onboard applications. Expert Systems with Applications, 256, 124848.

[26] Li, Y., Zhang, W., Lv, S., Yu, J., Ge, D., Guo, J., & Li, L. (2025). YOLOv11-CAFM model in ground penetrating radar image for pavement distress detection and optimization study. Construction and Building Materials, 485, 141907.

[27] Tetila, E. C., Junior, G. W., Higa, G. T. H., da Costa, A. B., Amorim, W. P., Pistori, H., & Barbedo, J. G. A. (2025). Deep learning models for detection and recognition of weed species in corn crop. Crop Protection, 107237.

[28] He, L. H., Zhou, Y. Z., Liu, L., Zhang, Y. Q., & Ma, J. H. (2025). Research on the directional bounding box algorithm of YOLO11 in tailings pond identification. Measurement, 117674.

[29] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., ... & Wang, X. (2022, October). Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision* (pp. 1-21). Cham: Springer Nature Switzerland.