Transformer or CNN? Assessing Real-Time Detection Transformer and YOLOv8 for small UAS Safe and Autonomous Landing

Can Vu*[1], Mahyar Ghazanfari †[1] , Kevin Dong‡, Abenezer Taye§, Amin Tabrizian¶ and Peng Wei‖
George Washington University, Washington, DC, 20052

Small Unmanned Aircraft Systems (UAS) are increasingly used in commercial package delivery, requiring robust object detection capabilities for safe and reliable landing in dynamic, off-nominal environments. Deep learning based computer vision models have shown promise in achieving high-accuracy object detection and show promise as key components to the UAS autonomy stack. In recent years, transformer architectures have emerged as powerful alternatives to convolutional models, offering strong representation capabilities for object detection tasks. This paper compares two state-of-the-art object detection models for aerial-view object detection: the convolutional neural network (CNN) YOLOv8 and Real-Time Detection Transformer (RT-DETR) models. Both models are trained and tested on the VisDrone aerial imagery dataset, which provides relevant data for detecting variably scaled and dense urban objects commonly encountered during UAS descent and landing. Performance is assessed using standard computer vision detection accuracy metrics and inference latency to assess real-time feasibility. To further evaluate the reliability of each model's predictions, we apply the conformal prediction framework, enabling statistically valid estimates of classification uncertainty. Our results provide insight into the trade-offs between detection performance, inference speed, and uncertainty calibration, offering guidance for selecting appropriate object detection models in safety-critical UAS landing applications.

I. Introduction

Package delivery using small unmanned aerial systems (UAS) is experiencing accelerated innovation and is closer to large scale commercialization. The Federal Aviation Administration (FAA) recently granted approval for companies such as Zipline and Wing Aviation to operate commercial UAS in the Dallas-Fort Worth (DFW) area without requiring visual observers, enabling beyond-visual line of sight (BVLOS) operations [1–3]. This authorization marked a key turning point in the regulatory landscape aimed at the integration of routine small UAS deliveries safely into the low-altitude airspace. Leading UAS package delivery companies have developed methods to safely manage UAS during delivery operations using Unmanned Aircraft System Traffic Management (UTM) services [2]. This development represents a significant step toward enabling routine BVLOS operations in shared airspace, driven by advancements in aircraft autonomy and UTM technologies.

Wing's approach focuses on the direct delivery of lightweight packages from businesses to consumers. The company, a subsidiary of Alphabet, has developed a fleet of autonomous drones designed to transport small items quickly and efficiently, catering to on-demand needs [4]. As of October 2024, the company serves 26 cities and towns around the DFW metroplex, with an average flight time from stores to customers of 3 minutes and 24 seconds [5]. In contrast, Zipline is renowned for its pioneering work in medical supply delivery, particularly in African nations such as Rwanda and Ghana. The company operates one of the world's largest autonomous small UAS delivery networks, with service areas encompassing the United States, Japan and other countries. Zipline's delivery system is versatile, serving not only healthcare but also retail and logistics sectors by integrating with existing e-commerce and parcel carrier services [3].

*Ph.D. Student, Department of Mechanical and Aerospace Engineering, George Washington University, AIAA Student Member.
†Ph.D. Student, Department of Mechanical and Aerospace Engineering, George Washington University, AIAA Student Member.
‡Research Intern, George Washington University.
§Ph.D. Student, Department of Mechanical and Aerospace Engineering, George Washington University, AIAA Student Member.
¶Ph.D. Student, Department of Computer Science, George Washington University, AIAA Student Member.
‖Associate Professor, Department of Mechanical and Aerospace Engineering, George Washington University, AIAA Associate Fellow.
[1]These authors contributed equally to this work and share first authorship.

As BVLOS small UAS delivery operations become increasingly prevalent, ensuring the safety and precision of UAS landings becomes increasingly important. Fast and accurate vision-based landing systems will play a crucial role in enabling reliable and scalable autonomous deliveries, especially in complex environments where GPS signal may be denied. To achieve a higher level of autonomy during an off-nominal landing, a UAS needs to recognize the landing scene configuration with moving or static objects near the designated landing pad (such as people, pets, cars, or bicycles) with high accuracy in real-time.

With this motivation, this paper investigates the Real-Time Detection Transformer (RT-DETR) as an alternative for small UAS landing scene object awareness, reportedly outperforming previous state-of-the-art vision models based on convolutional neural networks (CNN) on the COCO 2017 dataset [6]. The RT-DETR performance is assessed in terms of inference speed, mean average precision (mAP), and robustness (compared to our currently used YOLOv8 [7, 8] model) on our UAS flight test data to assess its feasibility in supporting real-time, safety-critical UAS landing operations. Furthermore, conformal prediction (CP) theory is applied to rigorously evaluate and quantify the classification uncertainty associated with each model.

II. Related Works

A. Transformer Architectures

Vaswani et al. [9] reported the first transformer architecture for sequence transduction modeling. To circumvent the sequential computational constraint of previous methods, an architecture based entirely on the Attention mechanism was developed and reached state-of-the-art performance on translations with only 12 hours of training and has since become the standard for natural language processing applications. Its uses in vision tasks were limited initially. Efforts to integrate the attention mechanisms ubiquitous in transformers with CNNs were investigated in [10, 11]. Carion et al. [12] introduced the Detection Transformer (DETR) to address the negative impact to speed and accuracy of non-maximum suppression (NMS) techniques typically found in modern object detection models. Dosovitskiy et al. [13] showed that reliance on CNN architectures were not necessary, and a pure transformer model, the Vision Transformer (ViT), applied to sequences of images can perform well on image classification with relatively lower pre-training costs. Liu et al. [14] developed the Swin Transformer, a vision transformer whose hierarchical feature representation is calculated using a sliding window approach, capable as a general purpose backbone across computer vision applications.

Further research was conducted to apply the transformer-based models to object detection. One study by Zhang [15] reported that the ViT consistently outperform CNN-based models for object detection and classification tasks; however, using the ViT requires more training data, computational power, and more intricate structure design. Another paper from Zhang et al. [16] showed that elements from both YOLO and ViT can be combined into a hybrid model that outperforms previous YOLO models in terms of detection accuracy on the VisDrone-DET 2021 challenge. Song et al. [17] integrated the ViT with the DETR to build an efficient and effective object detector. In real-time applications, the Real-Time Detection Transformer (RT-DETR), introduced by Zhao et al. [6], is the first real-time, end-to-end transformer-based object detection model. The RT-DETR model achieved state-of-the-art performance on the COCO dataset [18] by introducing an efficient hybrid encoder and uncertainty-minimal query selection, offering new opportunities for real-time object detection.

B. Deep Learning for Object Awareness in UAS Landing Operations

Recent studies have investigated the use of transformers elements for object detection in UAS autonomous landing. One study by Yue et al. [19] reported a vision-based autonomous UAS landing model consisting of a YOLOv5 object detection method and Spatio-Temporal Transformer for landing on dynamic platforms under an assumption of no landmarks or global coordinates. Neves et al. [20] utilized a multimodal transformer-based DL detector that provides reliable positioning for precise autonomous landing. Vazquez-Meza and Martinez-Carranza [21] presented a methodology for landing zone detection using ViT, which offers an architecture capable of capturing spatial relations among visual data. It was reported that the pre-trained ViT excel in scenarios where understanding global dependencies and context is critical but require more training data to reach comparable performance to CNNs.

As part of our group's drive toward safe autonomous UAS landing, Yang et al. [22] developed a deep learning (DL) based computer vision model using RetinaNet [23] for pedestrian detection by small UAS during dynamic aerial missions. Their work demonstrated high accuracy in detecting pedestrians and vehicles in a congested environment

despite the relatively small size of the objects from a bird's eye view. Zhao et al. [24] extended this work by comparing RetinaNet and YOLOv5 [25] for real-time landing pad and object detection, including an implementation of a safety checking algorithm using the object's bounding box and landing pad coordinates. It was reported that YOLOv5 and RetinaNet yielded similar accuracy; however, YOLOv5 was selected for its higher inference speed as the object detector for this work. ArUco markers were also introduced for more precise landing pad identification. The YOLOv5 model was pre-trained on the COCO dataset and fine-tuned on the VisDrone [26] dataset. The YOLOv5 model was trained on a workstation running Ubuntu 20.04 with an Intel Xeon W-3245 @ 4.60 GHz (32 cores), 128GB RAM, and an RTX 3080 GPU (10GB) and a trimmed version of the model was deployed on an Nvidia Jetson Xavier NX for model evaluation. It was shown that the YOLOv5 model could successfully detect the landing pad and pedestrians through real-world data testing, and the integrated safety checking algorithm successfully checked for the safety of the landing zone by ensuring any objects crossing into the landing pad area will flag a signal to the UAS that the area is unsafe.

C. Conformal Prediction

Gammerman et al. [27] proposed the original formulation of CP, introducing a transductive learning framework that outputs both predictions and associated confidence measures. This was further refined by Saunders et al. [28], who introduced formal definitions of confidence and credibility, establishing a practical method for uncertainty estimation in classification tasks using support vector machines. These works laid the groundwork for modern conformal prediction techniques used across machine learning applications to provide statistically valid guarantees about model reliability.

In recent years, CP has been extended to DL-based object detection tasks where uncertainty quantification is critical for safe, real-world deployment. de Grancey et al. [29] investigated the use of conformal prediction for object detection and proposed CP to integrate probabilistic guarantees into ground-level pedestrian bounding box predictions. Andeol et al. [30] applied conformal risk control and CP to the railway signal detection domain. By leveraging CP on a custom vision dataset collected from the train operator's point of view, the authors developed a robust framework that yields formally guaranteed uncertainty bounds. Their findings show that CP methods can be adapted to constrained environments where model reliability is essential. Saboury and Uyguroglu [31] applied CP to real-time visual anomaly detection in dynamic indoor environments using mobile robots. Their system incorporates CP into deep autoencoder architectures to detect anomalies while providing interpretable uncertainty measures for safety-aware robotic navigation. In the aerial and satellite imaging domain, Copley et al. [32] investigated the use of conformal prediction to quantify uncertainty in object detection models applied to Intelligence, Surveillance, and Reconnaissance (ISR) scenarios. Their approach estimates both detection and localization uncertainty in YOLO-based detectors and provides guarantees on bounding box coverage, but saw performance costs in bounding box precision and size.

III. Methods

A. Aerial-View Object Detection Dataset

For the task of UAS-view landing environment visual understanding, the DL models are trained on the large-scale benchmark VisDrone2019-DET dataset. This dataset was collected by the AISKYEYE [33] team at the Lab of Machine Learning and Data Mining at Tianjin University, China. The dataset consists of 288 video clips formed by 261,908 frames and 10,209 static images captured from various UAS-mounted camera covering various urban and rural locations, crowded and sparse environmental densities, object types, and lighting conditions. The images were collected from a variety of altitude and camera angles and manually annotated by the AISKYEYE team. The VisDrone dataset contains 2.6 million bounding boxes among ten object classes (Pedestrian, Person, Bicycle, Car, Van, Truck, Tricycle, Awning-Tricycle, Bus, and Motor). Figure 1 shows sample images from the training set of this dataset, showing the variation in lighting, perspective, and object density, available to train our models on diverse scene information.

B. Candidate Deep Learning Models for Aerial-View Object Detection

In this paper, the vision system of the UAS will be implemented via DL models for predicting the presence of objects (i.e. cars, pedestrians, bikes, etc.) around a landing site. The models predict the bounding boxes and confidence scores associated with the object detections. Driving toward high-accuracy, real-time autonomous UAS operation, two high performance object detection models are studied: YOLOv8 and RT-DETR.

(a) **(b)**

(c) **(d)**

Fig. 1: Example images from the training set of the VisDrone2019-DET dataset. The sample images cover a variety of scenes such as various object types, object densities, locations, and aerial angles and altitudes.

1. YOLOv8

The DL based object detection model selected for this work was the Ultralytics YOLOv8 model [7]. The YOLOv8 model was first released in 2023, offering cutting-edge speed and accuracy in object detection tasks at the time. An improvement on the YOLOv5 model, YOLOv8 introduces key innovations and enhancements to performance, efficiency, and ease of use. The base architecture of YOLOv8 (like YOLOv5) consists of a backbone, neck, and head structure. The enhanced CSPDarknet backbone [34] convolutional network extracts and processes a set of meaningful features from the input images. The neck uses the Path Aggregation Network [35] (PANet) structure, building on the Feature Pyramid Network, to perform feature fusion and contextual information integration on the extracted features from the backbone. This boosts the performance on small object detection. The head, featuring an anchor-free design, is the final layer outputting the bounding boxes and confidence scores. The full architecture of YOLOv8 is given in Figure 2. YOLOv8 optimizes a composite loss function that is the weighted sum of three loss terms and is given in Equation 1,

$$L_{\text{total}} = \lambda_{\text{box}} \cdot L_{\text{box}} + \lambda_{\text{cls}} \cdot L_{\text{cls}} + \lambda_{\text{dfl}} \cdot L_{\text{dfl}} \tag{1}$$

where L_{box} is the bounding box regression loss, L_{cls} is the binary cross-entropy classification loss, and L_{dfl} is the distribution focal loss (dfl). The weighting coefficients λ_{box}, λ_{cls}, and λ_{dfl} control the relative importance of each loss component. These values are tuned to balance localization accuracy, classification performance, and boundary precision.

While YOLOv8 offers high-speed and accuracy on object detection benchmarks, it has limitations in scenes with dense object clusters or partial occlusions. Moreover, its reliance on non-maximum suppression (NMS) can introduce latency and hyperparameter sensitivity. This work uses the lightweight YOLOv8s variant, which maintains the core architecture while reducing parameter count for faster inference.

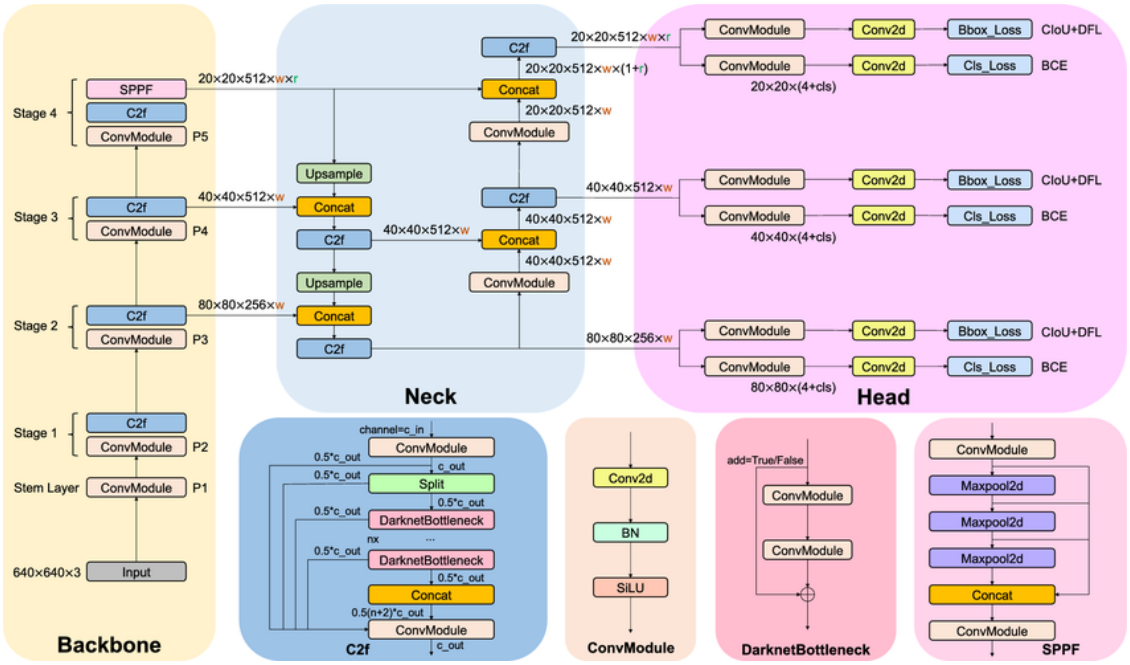


Fig. 2: A detailed system block diagram of the YOLOv8 object detection model showing the internal components of the backbone, head, and neck structures. [36].

2. Real-Time Detection Transformer

In contrast to YOLOv8's convolutional architecture, RT-DETR adopts a transformer-based design to achieve end-to-end object detection with competitive speed and accuracy. The high-level architecture of RT-DETR consists of a backbone, an efficient hybrid encoder, and a transformer decoder with auxiliary prediction heads [6]. Features from the last three stages of the backbone are processed by the encoder, which performs intra-scale feature interaction and cross-scale fusion to generate a compact set of multi-scale image features. RT-DETR introduces two key modules in its hybrid encoder to optimize for real-time performance: the Attention-based Intra-scale Feature Interaction (AIFI) and the CNN-based Cross-scale Feature Fusion (CCFF). Together, these components reduce computational overhead while maintaining a robust feature representation.

Current query selection schemes lead to a considerable level of uncertainty in the selected features, making for sub-optimal initializations in the decoder and reducing performance in the detector. To address this limitation, RT-DETR introduces an uncertainty-minimal query selection that constructs and optimizes the uncertainty to model the joint latent variable of encoder features, providing high-quality queries for the decoder. Specifically, the feature uncertainty $\mathcal{U}$ is defined as the discrepancy between the predicted distributions of localization $\mathcal{P}$ and classification C,

$$\mathcal{U}(\hat{X}) = \left\| \mathcal{P}(\hat{X}) - C(\hat{X}) \right\| \tag{2}$$

where $\hat{X} \in \mathbb{R}^D$ is the encoder feature. The loss function incorporates the feature uncertainty for gradient-based optimization. The loss function is a composite of the localization and classification loss shown in Equation 3

$$\mathcal{L}(\hat{X}, \hat{\mathcal{Y}}, \mathcal{Y}) = \mathcal{L}_{\text{box}}(\hat{b}, b) + \mathcal{L}_{\text{cls}}(\mathcal{U}(\hat{X}), \hat{c}, c) \tag{3}$$

where $\hat{\mathcal{Y}}$ and $\mathcal{Y}$ denote the prediction and ground truth, respectively, containing classification and bounding box values.

The complete RT-DETR pipeline is illustrated in Figure 3. The RT-DETR model was shown to achieve higher accuracy than previous state-of-the-art object detectors on the COCO benchmark, outperforming both YOLOv8 and DETR models. Its fully end-to-end design eliminates the need for post-processing steps like non-maximum suppression (NMS), simplifying deployment and reducing latency for practical application. The improvements to the hybrid encoder and introduction of the uncertainty-minimal query selection allows the detector to quickly processes multi-scale features, and improve the quality of initial object queries. Despite its strong performance on large and medium-sized objects, RT-DETR still underperformed on small object detection compared to specialized CNN-based detectors like YOLOv8.

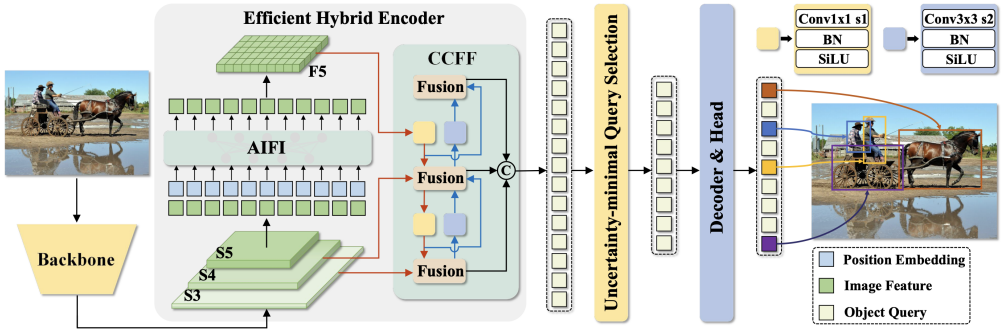


Fig. 3: A detailed system block diagram of the RT-DETR end-to-end object detection model. [6].

C. Conformal Prediction for Multi-Object Classification in UAS Aerial Imagery

Aerial imagery captured by Unmanned Aerial Systems (UAS) often contains multiple, densely packed, and variably scaled objects, making accurate classification under uncertainty a critical task. To address this, we adapt the Conformal Prediction (CP) [37]to quantify the uncertainty in class label predictions produced by a DL-based object detector. Conformal prediction is a distribution-free framework for generating statistically valid prediction sets or intervals for supervised learning tasks. Given a trained model $\hat{f}$ and a small calibration dataset $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ of i.i.d. samples from the same distribution as the test data, conformal prediction forms a prediction set $C(X_{\text{test}})$ for a new input X_{test} that satisfies the marginal coverage guarantee:

$$1 - \alpha \leq \mathbb{P}(Y_{\text{test}} \in C(X_{\text{test}})) \leq 1 - \alpha + \frac{1}{n+1}, \tag{4}$$

where $\alpha \in (0, 1)$ is the user-specified error level and n is the size of the calibration set. In classification, a common score function is the softmax-based conformal score:

$$s_i = 1 - \hat{f}(X_i)_{Y_i}, \tag{5}$$

which measures the model's lack of confidence in the true label. The threshold $\hat{q}$ is chosen as the $\lceil (n+1)(1-\alpha) \rceil / n$ quantile of the calibration scores. The prediction set is then defined as:

$$C(X_{\text{test}}) = \left\{ y \in \{1, \ldots, K\} : \hat{f}(X_{\text{test}})_y \geq 1 - \hat{q} \right\}. \tag{6}$$

Conformal Prediction (CP) yields prediction sets that adapt to model uncertainty: sets are smaller when the model is confident in its prediction and larger when uncertainty is higher. The procedure used to generate class prediction sets via CP for both YOLOv8 and RT-DETR is summarized below.

1) **Calibration via Validation Set:** The validation split of VisDrone is reserved as the calibration set. For each prediction in the calibration data, we compute a nonconformity score that reflects the model's uncertainty. In our case, this is typically defined as $s = 1 - \hat{p}$, where $\hat{p}$ is the predicted confidence score for the ground-truth class.
2) **Choice of Confidence Level:** The desired error tolerance $\alpha \in [0, 1]$ is fixed, which corresponds to a target confidence level of $1 - \alpha$ for the prediction sets.
3) **Quantile Computation:** We then compute the empirical quantile $\hat{q}$ of the nonconformity scores $s_1, \ldots, s_n$ at level $\lceil (n+1)(1-\alpha) \rceil / n$, where n is the number of calibration examples. This quantile serves as the threshold used to determine which detections are included in the conformal prediction sets.
4) **Inference with Conformal Sets:** With $\hat{q}$ in hand, we evaluate the model on the VisDrone test set. For each test-time detection, we retain only those predictions whose scores exceed $1 - \hat{q}$. This yields a conformal prediction set with theoretical marginal coverage of at least $1 - \alpha$. The resulting histograms summarize the prediction set sizes and coverage properties on the test set.
5) **Visual Predictions on Unseen Data:** Finally, we demonstrate the practical effectiveness of the calibrated model by applying it to previously captured images. The prediction sets visualized on these images qualitatively highlight the model's uncertainty in a per-instance manner.

IV. Experimental Setup

A. YOLOv8 and RT-DETR Model Training Hyperparameters

The YOLOv8 and RT-DETR models initialized with COCO pre-trained weights before being fine-tuned with the VisDrone train and validation datasets, respectively, on our workstation, which consists of an AMD Ryzen Threadripper PRO 3975WX (32-cores) CPU and two GeForce RTX 3090 GPUs. To maintain a fair comparison, the hyperparameters for training were kept the same in both training sessions, with the exception of the batch size. For YOLOv8, we train with a batch size of 8, while RT-DETR was trained with a batch size of 2 to accommodate GPU memory constraints. Early stopping based on validation performance was used to prevent overfitting with a patience of 15 epochs. Input images were resized to 720×1280 pixels. Optimization was conducted using the AdamW optimizer with an initial learning rate of 0.0001, and cosine learning rate scheduling (cos_lr=True) was applied with a final learning rate factor (lrf = 0.01). A warm-up phase was applied over the first 5 epochs.

To enhance generalization, a set of data augmentations was employed, including mosaic augmentation (mosaic = 1.0), mixup (mixup = 0.2), and random transformations (auto_augment = randaugment). Additional augmentations included horizontal flipping (fliplr = 0.5), HSV color perturbations, random translation (translate = 0.1), scaling (scale = 0.5), and random erasing with a probability of 0.4. Mixed-precision training (AMP) was enabled, and deterministic behavior was enforced by setting the random seed to 1. The initial layers of each model were partially frozen during early training (freeze = 10) to retain low-level features from the backbone.

B. UAS Landing Flight Test Data

The test dataset comprises three six-minute videos simulating the vertical descent of an unmanned aerial system (UAS) onto a designated landing site under clear daylight conditions. The landing target consists of a 5-foot diameter Hoodman Drone Landing Pad and includes four ArUco markers to further define the delivery zone. These markers assist in localizing the UAS within the immediate vicinity above the pad before descent. All video footage was captured using a DJI Mini 3 drone at various altitudes, providing a top-down aerial view of the landing area. The videos were recorded at a frame rate of 30 frames per second with a resolution of 1920×1080 pixels (Full HD). Each video includes a variety of common real-world objects that might be encountered during a typical UAS landing scenario, such as pedestrians and parked or moving vehicles. This test dataset is used to evaluate both object detection models, YOLOv8 and RT-DETR, on their standard point predictions, outputted by the Ultralytics framework, as well as their set prediction outputs from our CP method.

V. Results and Discussion

A. Training and Evaluation of YOLOv8 and RT-DETR: Accuracy and Processing Speeds on VisDrone

To compare the two models, the training metrics for both models are plotted alongside their normalized confusion matrices. The training hyperparameters of each model are described in Section IV.A. A higher mAP@50 indicates a more accurate model on "easy" detections. Similarly, mAP@50-95 evaluates the accuracy of the model by calculating the average mAP across a range of IoU thresholds from 0.5 to 0.95, giving a more comprehensive view of the model's detection performance across varying levels of localization difficulty. A higher mAP indicates that the object detector made more precise and consistent predictions.

The training results of the YOLOv8 and RT-DETR models, along with their respective confusion matrices, are presented in Figures 4 and 5. As shown in Figures 4a and 5a, both models demonstrate consistent reductions in training and validation losses over the course of training, accompanied by steady improvements in mAP@0.5 and mAP@0.5:0.95. While YOLOv8 converges more quickly in the early epochs, RT-DETR continues to improve over time and ultimately surpasses YOLOv8 in detection performance. The trained YOLOv8 model achieved a maximum mAP@0.5 of 0.4914 and a mAP@0.5:0.95 of 0.3073. In comparison, the RT-DETR model achieved higher performance, with a mAP@0.5 of 0.5270 and a mAP@0.5:0.95 of 0.3371. These results indicate that RT-DETR offers superior object detection accuracy across the full range of IoU thresholds when applied to the aerial-view object detection domain. This suggests that RT-DETR provides more precise and consistent predictions than YOLOv8 when presented with a dataset containing complex urban scenes where object scale, occlusion, and weather conditions may vary significantly.

Furthermore, a comparison of the confusion matrices of the YOLOv8 and RT-DETR models, shown in Figures 4b and 5b respectively, demonstrates that the RT-DETR model achieves higher classification accuracy across nearly all

Model	Pre-processing (ms/frame)	Inference (ms/frame)	Post-processing (ms/frame)	Total (ms/frame)
YOLOv8	9.2	186.4	5.8	201.4
RT-DETR	9.5	1252.3	0.5	1262.3

Table 1: Average processing speed breakdown of YOLOv8 and RT-DETR. The average time (in milliseconds) spent on pre-processing, inference, and post-processing per frame is summed to compute the total runtime of the object detection pipeline.

object categories. This is evident from the stronger diagonal intensities in the RT-DETR matrix, indicating a higher proportion of correct predictions per class. RT-DETR shows improved performance in distinguishing between vehicle classes (van, truck, and bus) and pedestrian-related classes (pedestrian and people), which are more prone to confusion. While both models exhibit some misclassification among similar object types, such as between different types of vehicles or between pedestrians and background, RT-DETR consistently produces fewer off-diagonal errors. This suggests that RT-DETR is better calibrated and more robust in multi-class aerial object detection under the VisDrone test conditions, making RT-DETR the better choice for object awareness in the UAS package delivery landing scenario.

To assess the feasibility of real-time deployment, the processing speeds of YOLOv8 and RT-DETR were evaluated. The end-to-end processing speed, measured in milliseconds (ms) per frame, is defined as the sum of the pre-processing, inference, and post-processing times per frame. The average duration of each stage is reported in Table 1. Both models exhibit comparable pre-processing speeds, with YOLOv8 being marginally faster. However, RT-DETR shows a significantly longer inference speed due to its more complex and computationally intensive transformer-based architecture. The extended inference time observed in RT-DETR can be attributed to its transformer-based architecture, which introduces greater computational complexity and a larger parameter count (approximately 42 million) relative to YOLOv8 (approximately 11.2 million). Conversely, RT-DETR benefits from a substantially lower post-processing time due to the removal of NMS. This architectural difference indicates an operational trade-off: while RT-DETR incurs approximately 6.7 times higher inference latency than YOLOv8, it achieves a post-processing speed that is 11.6 times faster. The resulting total processing speed shows substantially slower end-to-end processing speed for RT-DETR (1.2623 seconds per frame) compared to YOLOv8 (0.2014 seconds per frame), limiting its practicality for real-time applications without further optimizations.

B. YOLOv8 and RT-DETR Prediction Performance on Real-World UAS Flight Data

In real-world UAS package deliveries, variations in landing conditions, such as changes in lighting or object occlusions, can introduce uncertainty into vision-based object detection. Assessing this uncertainty is critical for developing safe and autonomous landing systems. To address this, we implemented a CP approach (see Section III.C) to quantify the classification uncertainty of the object detectors examined in this paper. Rather than making *point* predictions using the object detection models, the CP method generates statistically valid *set* predictions that contain the correct label with marginal probability coverage, i.e. the correct label is guaranteed to be in the prediction set with probability almost exactly $1 - \alpha$ [37]. In the experiments, the point predictions are made via the ultralytics framework, while the set predictions are the output of the CP approach. Note that the discussion of these results focus on the classification aspect of the object detection predictions, not the localization.

Figure 6 presents two frames with object detection predictions from the UAS flight test dataset described in Section IV.B. In this scenario, the UAS hovers at an altitude of approximately 20 meters while observing a landing environment that includes both static and moving objects and a landing pad. Figure 6a shows the point predictions obtained by running inference with the Ultralytics framework. In this mode, the DL model will assign at most one label to each detected object as well as outputs the mAP@50 and mAP@50-95 across the entire dataset. Inference using the Ultralytics API was also performed with the RT-DETR model on the same dataset, though a sample frame is not shown. The YOLOv8 model obtained a mAP@50 of 0.9690 and mAP@50-95 of 0.7931. Similarly, the RT-DETR model obtained a mAP@50 of 0.9844 and mAP@50-95 of 0.8400. Based on these values, RT-DETR marginally outperforms YOLOv8 in both mAP@50 and mAP@50-95, indicating that RT-DETR is more effective at distinguishing between object categories in challenging visual conditions such as occlusion and scale variation. Although mAP reflects the overall detection performance on the flight test, it does not account for the reliability or uncertainty of the models' predictions, factors that are crucial for ensuring safe operation in real-world UAS landing settings.

RT-DETR

α	total sets	mean set size	coverage (%)
0.1	50675	1.28	90.1
0.05	53282	1.44	94.7
0.025	54708	1.81	97.3
0.01	55648	2.47	98.9

Total detections: 56253

YOLOv8

α	total sets	mean set size	coverage (%)
0.1	37657	1.07	87.8
0.05	39930	1.35	93.2
0.025	41284	1.75	96.3
0.01	42220	2.41	98.5

Total detections: 42850

Table 2: Tables for the mean class prediction set length and total number of sets containing the true class label for varying values of error tolerance α for each DL model.

Unlike the inference performed in Figure 6a, CP allows the object detector to predict a set of class labels for each detected object. This behavior is illustrated in Figure 6b, where an object near a car is assigned the class set {pedestrian, people}. This set indicates that the predicted confidence scores for both the "pedestrian" and "people" classes exceeded the conformal threshold, qualifying them for inclusion in the prediction set. For this object, the correct label for this object is pedestrian and is contained in the class prediction set for that object. Class prediction sets are generated for every object detected by the deep learning model in a frame. Depending on the user-specified confidence level, these sets may include any or all of the ten object classes defined in the VisDrone dataset. The CP method is used to generate class set size distributions across the entire VisDrone test dataset, with the VisDrone validation set serving as the calibration data.

Figure 7 presents the estimated distribution of prediction set sizes that contained the true class label for $\alpha = \{0.1,\ 0.05,\ 0.025,\ 0.01\}$, corresponding to confidence levels of $1 - \alpha = \{90\%,\ 95\%,\ 97.5\%,\ 99\%\}$, respectively. The confidence level $1 - \alpha$ corresponds to the expected proportion of prediction sets that contain the true class label on unseen data. The histograms are generated by performing CP on the VisDrone test dataset and plotting the frequency of the predicted class set sizes for each DL model studied in the paper. Varying the error tolerance α results in shifts in the distribution of prediction set sizes. At $\alpha = 0.1$ (corresponding to 90% confidence), a large proportion of the class prediction sets are of size one. This is because the conformal threshold for inclusion is relatively high; only the most confidently predicted label is included in the set. As α decreases, the threshold becomes less strict, leading to a higher frequency of larger prediction sets in order to maintain the desired coverage guarantee. This indicates that the model is often confident enough to make a single-label prediction while still satisfying the desired coverage level. This behavior suggests a trade-off between the set size and coverage. Smaller prediction sets are generally more informative, and their prevalence at higher α values suggests efficient uncertainty estimation.

Key statistics of the estimated distribution of class prediction sets are shown in Table 2. For each value of α, the table reports the total number of predicted class sets that contain the true label, the average size of the prediction sets, and the corresponding empirical coverage. Additionally, the total number of valid detections made by each model is reported below each respective table. The empirical coverage is defined as the ratio of the total prediction sets containing the true label to the total number of valid predictions made by the object detection model. As the value of α decreases, the average prediction set size and empirical coverage increase in response. This corroborates the behavior previously observed in Figure 7, where the frequency of prediction set sizes greater than one increases as the conformal confidence threshold increases (or α decreases).

The empirical coverage achieved by RT-DETR closely aligns with the target confidence levels across all values of α. At $\alpha = 0.1$, the coverage slightly exceeds the desired threshold, and for smaller values of α, the coverage remains well-matched to the expected confidence levels. In contrast, YOLOv8 consistently underperforms with respect to coverage, falling below the target threshold for all evaluated values of α (although deviations from the target confidence values are relatively small). One possible reason for the discrepancy between the empirical coverage and the nominal confidence level is the underlying assumption of exchangeability. For the coverage guarantee in Equation 4 to hold, the calibration and test data must be exchangeable, meaning they come from the same distribution and are similarly structured but not necessarily independent and identically distributed (i.i.d.). Although the VisDrone validation and test sets are drawn from the same dataset and are similar in structure, there are no formal guarantees that they are exchangeable. As a result, the theoretical coverage guarantees provided by CP may not hold exactly in practice. Despite the lack of formal guarantees on exchangeability, the empirical coverage can still provide meaningful insight into the

uncertainty associated with the predictions made by the deep learning models. These results suggest that RT-DETR achieves better statistical coverage under the conformal prediction framework than YOLOv8, indicating that RT-DETR may be better suited for safety-critical or uncertainty-aware applications, such as safety-critical autonomous UAS landing operations.

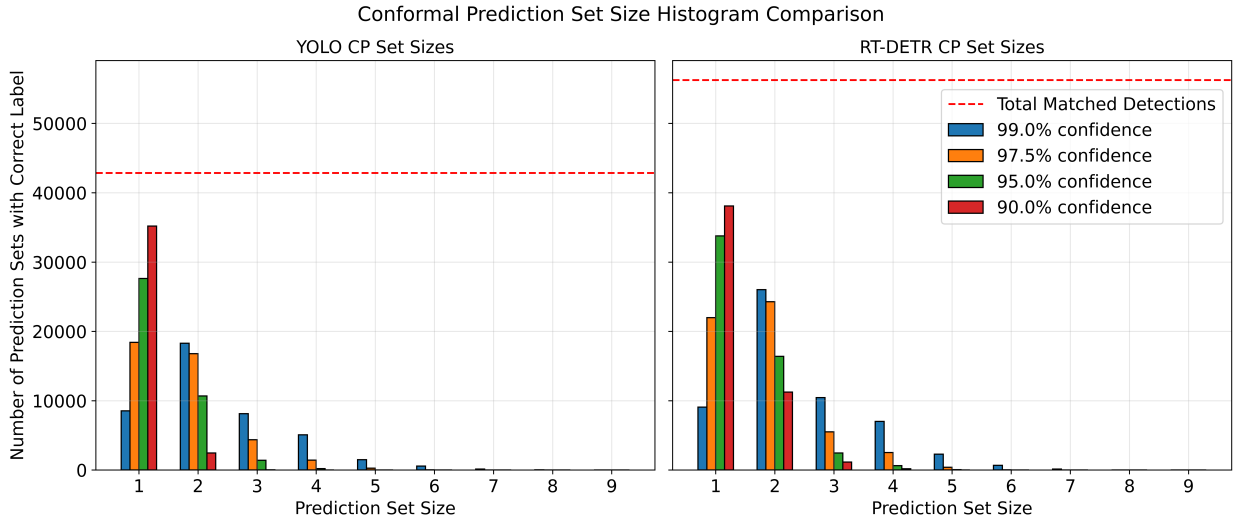


Fig. 7: Estimated distributions of the class prediction set sizes on the VisDrone test dataset. Four values of α were specified to observe the changes to the prediction set size distribution for each model. As α decreases, the confidence level increases, resulting in larger prediction sets to maintain the specified coverage.

VI. Limitations

Due to geographical limitations, the real-world flight test dataset used in this study consists of a small number of short-duration videos captured under clear daylight conditions with relatively few objects present. This limitation limits the diversity of environmental conditions and scene complexity, reducing the complexities that may be encountered during real-world UAS landing operations such as low-light scenarios, adverse weather conditions, or dense urban clutter. Additionally, the VisDrone training dataset captures only 9 common object types, limiting the extent of the objects that the UAS can encounter and successfully detect during descent. Future efforts can focus on expanding both the training and test datasets to incorporate a wider range of conditions and objects that can be encountered to increase the safety and effectiveness of autonomous UAS landing.

Another limitation involves the application of Conformal Prediction (CP). In this study, CP was applied exclusively to the classification outputs of the object detectors, providing valid prediction sets over possible class labels; however, the bounding box predictions were not included in the uncertainty quantification process, where precise object awareness is essential for safe landing, navigation, and object avoidance. Accurate localization is critical for spatial reasoning and safe navigation. Future work in this area can extend the CP method to quantifying the localization uncertainty to further improve safety-critical decision-making in off-nominal landing conditions.

VII. Conclusion

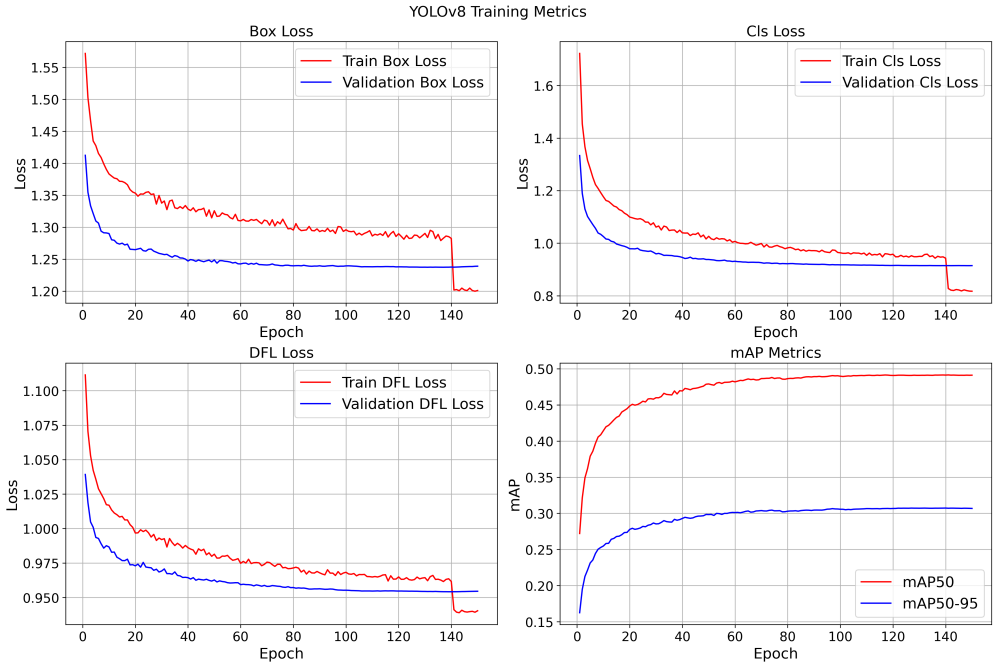
This study evaluated and compared two state-of-the-art DL object detection models, YOLOv8 and RT-DETR, for their suitability in supporting safe and autonomous landings of small Unmanned Aerial Systems (UAS). Using both the VisDrone aerial imagery dataset and real-world UAS landing test data, we trained and assessed the models on detection accuracy, inference speed, and classification uncertainty through the integration of CP. Our results show that while RT-DETR marginally outperforms YOLOv8 in terms of detection accuracy and classification precision, it comes at a significant cost to inference latency. With an average per-frame processing speed more than six times that of YOLOv8, RT-DETR's deployment on edge platforms remains limited without further model optimization or hardware acceleration.

In terms of uncertainty quantification using conformal prediction, RT-DETR demonstrated more reliable and

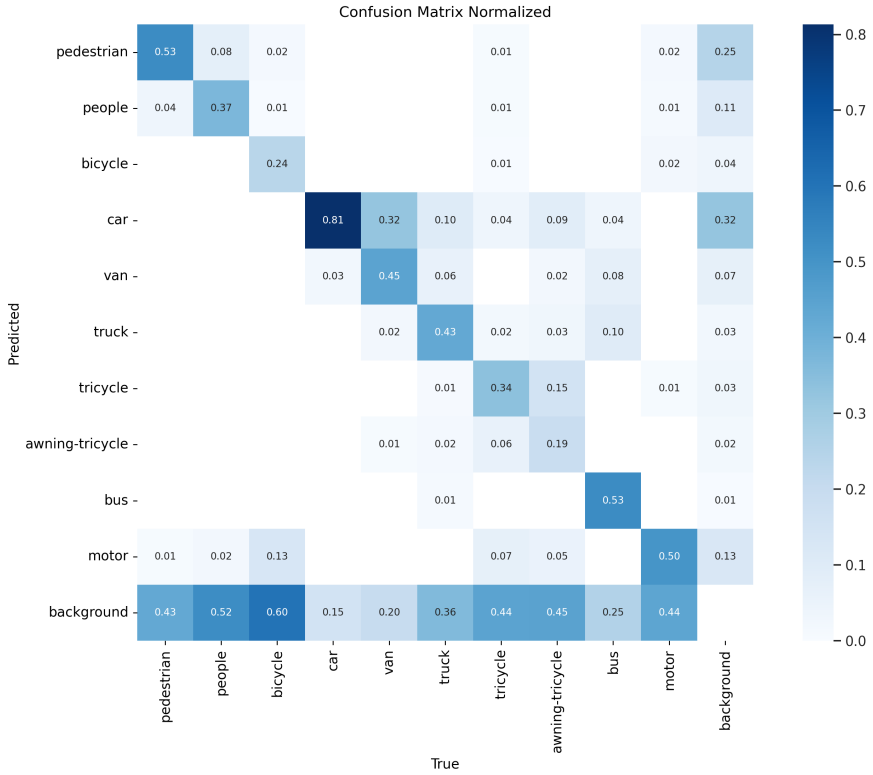
well-calibrated prediction sets compared to YOLOv8. Its empirical coverage closely matched the desired confidence levels across all tested values of α, indicating consistent and trustworthy uncertainty estimates. YOLOv8, while faster and more lightweight, exhibited slight undercoverage at each confidence level, suggesting less reliable calibration under the CP framework. These results imply that RT-DETR may be better suited for applications where uncertainty quantification is critical for safe, autonomous UAS landings, while YOLOv8 remains a practical choice when real-time performance is prioritized over statistical confidence guarantees.

Acknowledgments

We thank our colleagues at the FAA for their continued funding and support for our research.

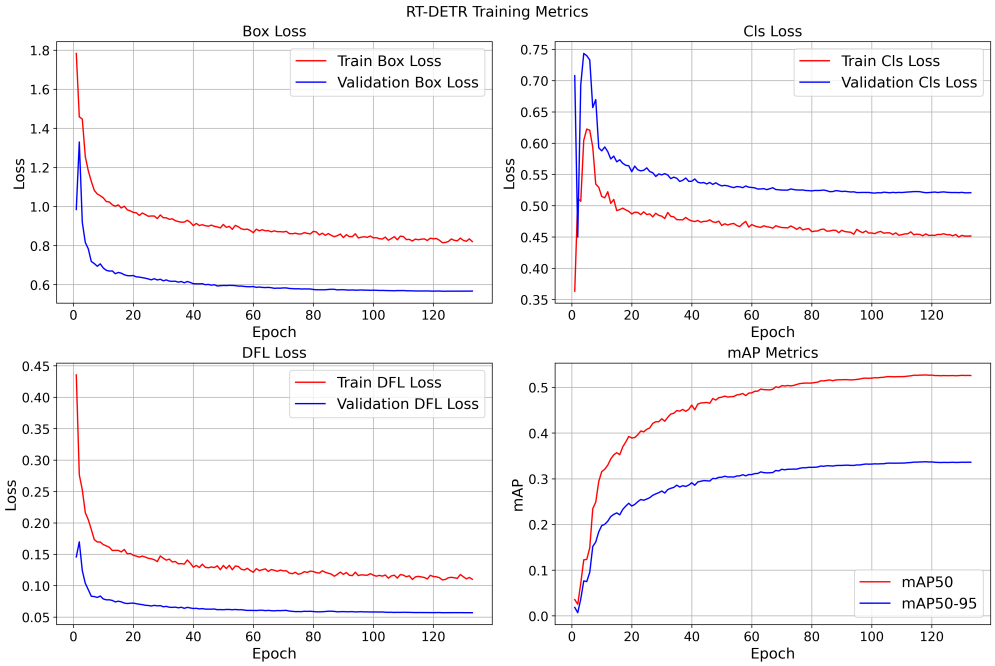


(a) Training plots for the YOLOv8 model. The first three plots show the training and validation loss curves, while the last plot shows mAP@0.5 and mAP@0.5:0.95.

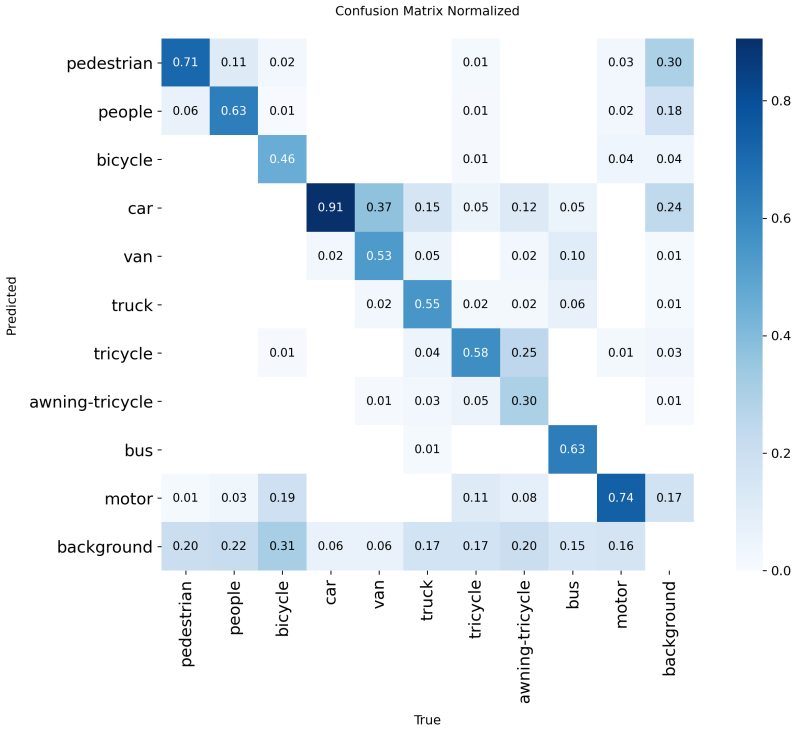


(b) Normalized confusion matrix for YOLOv8 model predictions. Most of the predictions are made along the diagonal, where correct predictions are shown. The off-diagonal values indicate the proportion of misclassifications made by the predictor.

Fig. 4: YOLOv8 plots for training metrics and confusion matrix on the VisDrone train dataset.



(a) Training plots for the RT-DETR model. The first three plots show the training and validation loss curves, while the last plot shows mAP@0.5 and mAP@0.5:0.95.



(b) Normalized confusion matrix for RT-DETR model predictions. Most of the predictions are made along the diagonal, where correct predictions are shown. The off-diagonal values indicate the proportion of misclassifications made by the predictor.

Fig. 5: RT-DETR plots for training metrics and confusion matrix on the VisDrone train dataset.

(a) Point predictions obtained from performing inference on a single frame from the UAS flight test data. All the objects were successfully detected with the correct classifications in the frame.

(b) Set predictions obtained from performing inference on a single frame from the UAS flight test data. Some of the objects have predictions that are the same as the point predictions of Figure 6a, while two of the detected objects are assigned classification sets from the CP.

Fig. 6: Two sample images containing the point and set predictions obtained from performing inference with Ultralytics (top) and CP (bottom) frameworks. These predictions are generated using the YOLOv8 DL model.

References

[1] Federal Aviation Administration, "Package Delivery by Drone (Part 135)," , 2023. URL https://www.faa.gov/uas/advanced_operations/package_delivery_drone, accessed: 2024-10-19.

[2] Federal Aviation Administration, "FAA Makes Drone History in Dallas Area," , 2024. URL https://www.faa.gov/newsroom/faa-makes-drone-history-dallas-area, accessed: 2024-10-19.

[3] Marquand, Y. L., "FAA authorises Zipline and Wing for BVLOS operations in Dallas," , 2023. URL https://www.revolution.aero/news/2024/07/30/faa-authorises-zipline-and-wing-for-bvlos-operations-in-dallas/, accessed: 2024-10-19.

[4] Singh, I., "Zipline, Wing enter new drone delivery era with historic FAA waiver," , 2024. URL https://dronedj.com/2024/07/30/zipline-wing-drone-delivery-dallas/, accessed: 2024-10-19.

[5] Nate Milner, "1 year and 26 cities later: Celebrating our DFW growth," , 2024. URL https://blog.wing.com/2024/10/1-year-and-26-cities-later-celebrating.html.

[6] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., and Chen, J., "DETRs Beat YOLOs on Real-time Object Detection," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16965–16974. https://doi.org/10.1109/CVPR52733.2024.01605.

[7] Jocher, G., Chaurasia, A., and Qiu, J., "Ultralytics YOLOv8," , 2023. URL https://github.com/ultralytics/ultralytics.

[8] Hussain, M., "YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision," , 07 2024. https://doi.org/10.48550/arXiv.2407.02988.

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I., "Attention is All you Need," *Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[10] Wang, X., Girshick, R., Gupta, A., and He, K., "Non-local Neural Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803. https://doi.org/10.1109/CVPR.2018.00813.

[11] Li, L., Xu, M., Wang, X., Jiang, L., and Liu, H., "Attention Based Glaucoma Detection: A Large-Scale Database and CNN Model," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S., "End-to-End Object Detection with Transformers," *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, Springer-Verlag, Berlin, Heidelberg, 2020, p. 213–229. https://doi.org/10.1007/978-3-030-58452-8_13, URL https://doi.org/10.1007/978-3-030-58452-8_13.

[13] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," , 2021. URL https://arxiv.org/abs/2010.11929.

[14] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986.

[15] Zhang, J., "Towards a High-Performance Object Detector: Insights from Drone Detection Using ViT and CNN-based Deep Learning Models," *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*, 2023, pp. 141–147. https://doi.org/10.1109/ICSECE58870.2023.10263514.

[16] Zhang, Z., Lu, X., Cao, G., Yang, Y., Jiao, L., and Liu, F., "ViT-YOLO:Transformer-Based YOLO for Object Detection," *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 2799–2808. https://doi.org/10.1109/ICCVW54120.2021.00314.

[17] Song, H., Sun, D., Chun, S., Jampani, V., Han, D., Heo, B., Kim, W., and Yang, M.-H., "ViDT: An Efficient and Effective Fully Transformer-based Object Detector," *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=w4cXZDDib1H.

[18] Lin, T. e. a., "Microsoft COCO: Common Objects in Context," *Computer Vision - ECCV 2014*, Springer, Cham, 2014. https://doi.org/https://doi.org/10.1007/978-3-319-10602-1_48, URL https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48#citeas.

[19] Yue, P., Xin, J., Mao, Z., Lu, Y., and Zhang, Y., "Vision-based Autonomous Landing of UAV on Dynamic Apron," *2023 IEEE International Conference on Unmanned Systems (ICUS)*, 2023, pp. 1613–1619. https://doi.org/10.1109/ICUS58632.2023.10318423.

[20] Neves, F. S., Branco, L. M., Pereira, M. I., Claro, R. M., and Pinto, A. M., "A Multimodal Learning-based Approach for Autonomous Landing of UAV," *2024 20th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2024, pp. 1–8. https://doi.org/10.1109/MESA61532.2024.10704866.

[21] Vazquez, V., and Martinez-Carranza, J., "Landing Zone Detection for MAVs using Depth Images and Vision Transformers," *15^{th} annual International Micro Air Vehicle Conference and Competition*, edited by T. Richardson, Bristol, United Kingdom, 2024, pp. 163–170. URL http://www.imavs.org/papers/2024/19.pdf, paper no. IMAV2024-19.

[22] Yang, X., Murphy, M., Brittain, M. W., and Wei, P., "Computer Vision for Small UAS Onboard Pedestrian Detection," *AIAA AVIATION 2020 FORUM*, American Institute of Aeronautics and Astronautics, VIRTUAL EVENT, 2020. https://doi.org/10.2514/6.2020-3270, URL https://arc.aiaa.org/doi/10.2514/6.2020-3270.

[23] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollar, P., "Focal Loss for Dense Object Detection," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[24] Zhao, Z., Lee, J., Li, Z., Park, C. H., and Wei, P., "Vision-based Perception with Safety Awareness for UAS Autonomous Landing," *AIAA SCITECH 2023 Forum*, American Institute of Aeronautics and Astronautics, National Harbor, MD & Online, 2023. https://doi.org/10.2514/6.2023-0126, URL https://arc.aiaa.org/doi/10.2514/6.2023-0126.

[25] imyhxy, "YOLOv5," , Accessed on 11 2024. URL https://github.com/ultralytics/yolov5.

[26] Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., and Ling, H., "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 44, No. 11, 2021, pp. 7380–7399.

[27] Gammerman, A., Vovk, V., and Vapnik, V., "Learning by transduction," *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, p. 148–155.

[28] Saunders, C., Gammerman, A., and Vovk, V., "Transduction with confidence and credibility," *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, p. 722–726.

[29] de Grancey, F., Adam, J.-L., Alecu, L., Gerchinovitz, S., Mamalet, F., and Vigouroux, D., "Object Detection with Probabilistic Guarantees: A Conformal Prediction Approach," *Computer Safety, Reliability, and Security. SAFECOMP 2022 Workshops*, edited by M. Trapp, E. Schoitsch, J. Guiochet, and F. Bitsch, Springer International Publishing, Cham, 2022, pp. 316–329.

[30] Andeol, L., Fel, T., de Grancey, F., and Mossina, L., "Confident Object Detection via Conformal Prediction and Conformal Risk Control: an Application to Railway Signaling," *Proceedings of the Twelfth Symposium on Conformal and Probabilistic Prediction with Applications*, Proceedings of Machine Learning Research, Vol. 204, edited by H. Papadopoulos, K. A. Nguyen, H. Boström, and L. Carlsson, PMLR, 2023, pp. 36–55. URL https://proceedings.mlr.press/v204/andeol23a.html.

[31] Saboury, A., and Uyguroglu, M. K., "Uncertainty-Aware Real-Time Visual Anomaly Detection With Conformal Prediction in Dynamic Indoor Environments," *IEEE Robotics and Automation Letters*, Vol. 10, No. 5, 2025, pp. 4468–4475. https://doi.org/10.1109/LRA.2025.3552318.

[32] Copley, V., Finlay, G., and Hiett, B., "The Uncertain Object: Application of Conformal Prediction to Aerial and Satellite Images," *Proceedings of the Thirteenth Symposium on Conformal and Probabilistic Prediction with Applications*, Proceedings of Machine Learning Research, Vol. 230, edited by S. Vantini, M. Fontana, A. Solari, H. Boström, and L. Carlsson, PMLR, 2024, pp. 73–89. URL https://proceedings.mlr.press/v230/copley24a.html.

[33] AISKYEYE, "Introduction - VISDRONE," Online, 2025. URL https://aiskyeye.com/home/, accessed: May 5, 2025.

[34] Khalili, B., and Smyth, A. W., "SOD-YOLOv8—Enhancing YOLOv8 for Small Object Detection in Aerial Imagery and Traffic Scenes," *Sensors*, Vol. 24, No. 19, 2024. https://doi.org/10.3390/s24196209, URL https://www.mdpi.com/1424-8220/24/19/6209.

[35] Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J., "Path Aggregation Network for Instance Segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768. https://doi.org/10.1109/CVPR.2018.00913.

[36] Ju, R.-Y., and Cai, W., "Fracture Detection in Pediatric Wrist Trauma X-ray Images Using YOLOv8 Algorithm," *Scientific Reports*, Vol. 13, 2023. https://doi.org/10.1038/s41598-023-47460-7.

[37] Angelopoulos, A. N., and Bates, S., "Conformal Prediction: A Gentle Introduction," *Found. Trends Mach. Learn.*, Vol. 16, No. 4, 2023, p. 494–591. https://doi.org/10.1561/2200000101, URL https://doi.org/10.1561/2200000101.