

Robust Vision-Based Runway Detection through Conformal Prediction and Conformal mAP

Alya Zouzou

ALYASLTD@GMAIL.COM

Paul Valery University, Montpellier III, France

Airbus, Toulouse, France

Léo Andéol

LEO.ANDEOL@MATH.UNIV-TOULOUSE.FR

Institut de Mathématiques de Toulouse, Toulouse, France

SNCF, Saint-Denis, France

Mélanie Ducoffe

MELANIE.DUCOFFE@AIRBUS.COM

Airbus, Toulouse, France

IRT Saint Exupéry, Toulouse, France

Ryma Boumazouza

RYMA.BOUMAZOUZA@AIRBUS.COM

Airbus, Toulouse, France

IRT Saint Exupéry, Toulouse, France

Editor: Khuong An Nguyen, Zhiyuan Luo, Harris Papadopoulos, Tuwe Löfström, Lars Carlsson and Henrik Boström

Abstract

We explore the use of conformal prediction to provide statistical uncertainty guarantees for runway detection in vision-based landing systems (VLS). Using fine-tuned YOLOv5 and YOLOv6 models on aerial imagery, we apply conformal prediction to quantify localization reliability under user-defined risk levels. We also introduce **Conformal mean Average Precision** (C-mAP), a novel metric aligning object detection performance with conformal guarantees. Our results show that conformal prediction can improve the reliability of runway detection by quantifying uncertainty in a statistically sound way, increasing safety on-board and paving the way for certification of ML system in the aerospace domain.

Keywords: Conformal Prediction, Average Precision, Object Detection, Aeronautics



Figure 1: Illustration of conformal prediction on the runway detection task. **Red: Ground Truth**, **Blue: YOLO Prediction**, **Green: Outer Conformalized Box**.

1. Introduction

The emergence of Machine Learning (ML), particularly deep learning and neural network (NN) models, has enabled transformative progress across a wide range of domains including transportation, healthcare, and finance. However, despite their impressive performance, neural networks exhibit inherent vulnerabilities, most notably their sensitivity to adversarial inputs and data distribution shifts, as highlighted in foundational work such as [Szegedy et al. \(2014\)](#). These weaknesses raise critical concerns when deploying ML systems in safety-critical contexts, where guarantees on reliability, robustness, and uncertainty quantification are paramount. One of the most stringent and high-stakes domains for ML deployment is aerospace. Within this field, vision-based landing systems (VLS) have emerged as a promising application of ML to enhance or potentially replace traditional landing aids. Aircraft landing is among the most complex and risk-prone phases of flight, currently supported by redundant human and technical systems—two-pilot operations, standardized communication protocols (e.g., ATIS), and ground-based navigation aids such as the Instrument Landing System (ILS) and GPS. As the industry moves toward single-pilot operations and increasing automation, ML-based systems must be able to provide reliable, interpretable, and certifiable support for tasks like visual runway detection and localization.

We explore recent developments in conformal prediction as a tool for uncertainty quantification (UQ) in object detection models applied to aerial imagery. These tools offer statistical guarantees on localization coverage and can help satisfy regulatory demands regarding AI in aerospace. Specifically, we focus on the core task of runway detection, which serves as a critical anchor for downstream aircraft pose estimation. Accurate detection of the runway is foundational—errors at this stage propagate through the pipeline and could compromise the entire landing guidance system.

Our contributions are as follows:

- We apply and evaluate conformal prediction with different margins over boxes (e.g., additive or multiplicative penalties) on multiple object detection architectures for the task of runway detection in aerial images.
- We introduce a novel evaluation metric, the conformal mean Average Precision (C-mAP), which quantifies the trade-off between accurate bounding box predictions and their coverage of ground-truth boxes at user-defined risk levels—thus aligning conformal prediction with established object detection benchmarks.
- We release our codebase ¹, trained models, and experimental protocols as an open-source repository to encourage reproducibility and further research in this critical area.

2. Object Detection Overview

Object detection refers to the task of identifying and localizing objects within an image, along with classifying them into predefined categories. Recent works in object detection show a great variety of models, mostly performed by neural networks, which can be categorized into two main types: **two-stage detectors** and **single-stage detectors**.

1. https://github.com/alyasltd/conformal_runway_detection

- **Two-stage detectors** first identify Regions of Interest (RoIs) in the image, and then classify those regions as containing or not objects (R-CNN, Mask R-CNN, [Bharati and Pramanik \(2020\)](#)).
- **Single-stage detectors**, such as the YOLO (You Only Look Once, [Terven et al. \(2023\)](#)) family, are known for their efficiency and trade-off between speed and accuracy. They directly predict bounding boxes and class probabilities for each object in a single pass through the network.

The goal of object detection can be formally defined as follows: given an image containing potential objects from a target set of classes, the object detector must localize each object and predict its class label. Localization is achieved through the use of bounding boxes. Each predicted bounding box is associated with the following information:

- **Localization vector**: a vector denoted as $\hat{\mathbf{b}} = [x_{min}, y_{min}, x_{max}, y_{max}]$, which are typically represented by their top-left and bottom-right coordinates.
- **Objectness score o** : A probability value between 0 and 1, representing the model's confidence that an object exists within the bounding box, regardless of its class label.
- **Class probability vector \mathbf{p}** : A vector containing class probabilities, where the highest value indicates the predicted class of the object, provided that the objectness score exceeds a certain threshold (indicating that the object exists).

From these two last sources of information, we compute the **confidence score**, defined as the product of the objectness score and the class probability. Although this score is sometimes used as a proxy for uncertainty estimation [Wenkel et al. \(2021\)](#), it lacks reliability—often over- or underestimating the true uncertainty—and provides no statistical guarantees.

In practice, object detectors can produce thousands of bounding boxes for a single image, many of which may significantly overlap. This overlap is typically quantified using the *Intersection over Union (IoU)* metric, defined as the area of the intersection divided by the area of the union of the predicted and ground truth bounding boxes: $\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$

To handle such redundancy, a post-processing step is applied, most commonly the Non-Maximum Suppression (NMS, [Neubeck and Van Gool \(2006\)](#)) algorithm, which removes overlapping bounding boxes. A second step consists in filtering out insufficiently confident boxes of the final sequence of prediction boxes.

We define an object detection model f as the composition of the object detector (after NMS) and confidence threshold. Formally, we start with a model f^{NMS} taking as input an image x and returning a sequence of fixed length N . These predictions are then filtered for (when two boxes shared an IoU over t , the least confident is disregarded). This results in a sequence of predictions of variable length N_x^{NMS} formally denoted by

$$f^{\text{NMS}} : x \rightarrow \left(\hat{\mathbf{b}}_i, o_i, \mathbf{p}_i \right)_{i=1}^{N_x^{\text{NMS}}},$$

where $\hat{\mathbf{b}}_i$ is the bounding box for the i -th prediction, o_i is the corresponding objectness score, \mathbf{p}_i is the vector of class probabilities.

Finally, all predictions with objectness scores above a threshold τ are preserved, $I(x) = \{i \in \{1, \dots, N_x^{\text{NMS}}\} : o_{[i]} \geq \tau\}$, where $o_{[i]}$ is the i -th largest objectness score in $f(x)$. The threshold τ is a predefined value from the filtering stage used to discard predictions with low objectness scores, ensuring that only detections with sufficiently high objectness score are considered. We then obtain the final predictions $f : x \rightarrow \{f^{\text{NMS}}(x)_{[k]}, k \in I(x)\}$.

2.1. Model Evaluation: Mean Average Precision (mAP)

The performance of object detection models is commonly evaluated using the **mean Average Precision (mAP)** metric, which is the mean of the **Average Precision (AP)** values computed across all object classes.

Consider the set of predicted bounding boxes $\{\hat{\mathbf{b}}_i^x\}$ output by the object detector, and the ground truth annotations be $\{(\mathbf{b}_j^x, c_j^x)\}_j$, where \mathbf{b}_j^x is a ground truth box and c_j^x its corresponding class label. Denoting C as the total number of classes, mAP is defined as:

$$\text{mAP} = \frac{1}{C} \sum_c AP \left(\{\hat{\mathbf{b}}_i^x\}_{\hat{c}_i^x=c}, \{\mathbf{b}_j^x\}_{c_j^x=c} \right),$$

where $\hat{c}_i^x = \arg \max_k \mathbf{p}_i^x(k)$.

We now focus on the definition of **Average Precision (AP)**. To compute AP, predicted bounding boxes are matched per image to ground truth boxes using a similarity measure, typically the IoU. A predicted box is considered a **true positive** if its IoU with a ground truth box exceeds a predefined threshold and the class label matches; otherwise, it is a **false positive**. Ground truth boxes not matched to any prediction are considered **false negatives**. Predicted boxes are ranked from highest to lowest confidence score (excluding those below a minimum confidence threshold), and this ranked list is used to construct a **precision-recall curve**. Recall that:

- **Precision** is the cumulative ratio of true positives to the sum of true positives and false positives: Precision : $p = \frac{\text{TP}}{\text{TP} + \text{FP}}$
- **Recall** is the cumulative ratio of true positives to the sum of true positives and false negatives: Recall : $r = \frac{\text{TP}}{\text{TP} + \text{FN}}$

3. Related Work

3.1. Uncertainty Quantification for Object Detection

Uncertainty quantification (UQ) is essential for object detection, especially in safety-critical applications where erroneous predictions can have significant consequences. Several methods have been proposed to estimate uncertainty, but many involve trade-offs that limit their practical use.

One common approach is Deep Ensembles, which estimates epistemic uncertainty by training multiple models and measuring the variance in their predictions. This technique has shown effectiveness in tasks such as robotic perception [Lyu et al. \(2020\)](#), but its major drawback is the computational cost associated with training several models, making it difficult to scale, particularly in real-time or large-scale applications.

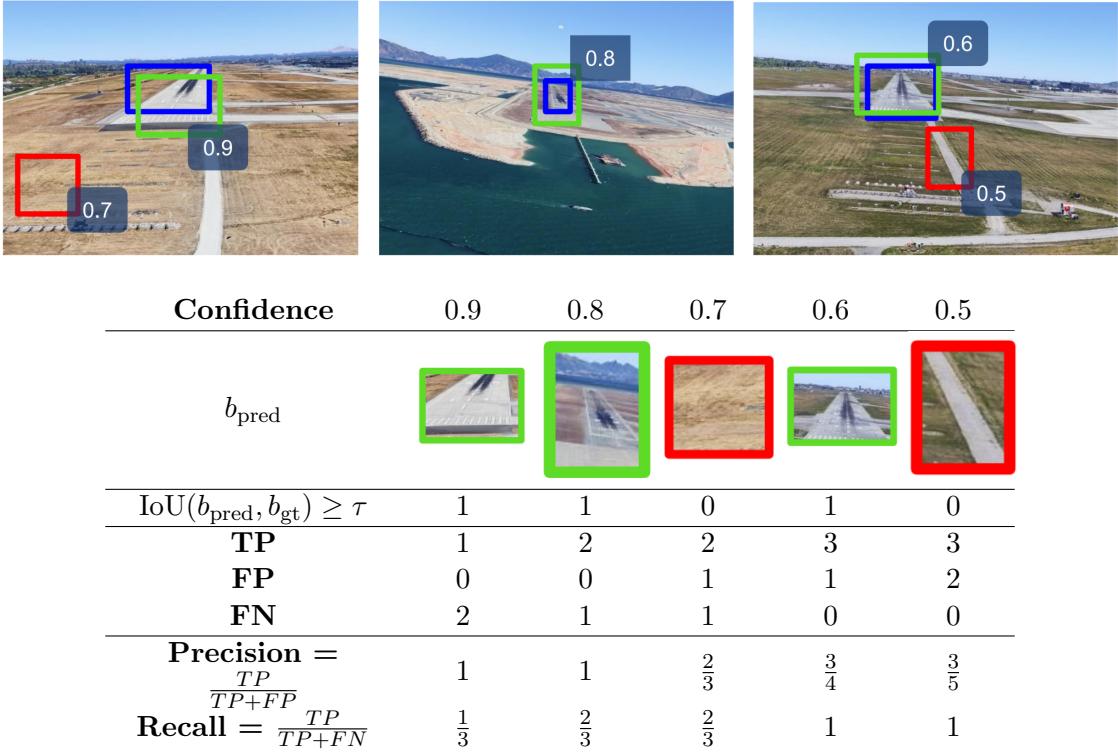


Figure 2: Computation of mAP step-by-step on an example from the LARD dataset. The top image shows the visual setup, while the table illustrates how predictions at different confidence levels affect precision and recall.

Another popular method is Dropout Sampling, which introduces stochasticity during inference to approximate Bayesian inference. This method has been shown to improve robustness under open-set conditions and enhance the model’s ability to quantify label uncertainty [Miller et al. \(2018\)](#). However, like Deep Ensembles, it requires modifications to the model and training process, which can be challenging for certain applications, especially when real-time performance is crucial. A more efficient technique, Monte Carlo DropBlock [Yelleni et al. \(2024\)](#), applies dropout during both training and testing phases, which improves generalization and calibration. While promising, it still relies on stochastic processes like Monte Carlo Dropout, which can lead to increased inference times, limiting its applicability in time-sensitive scenarios. BayesOD [Harakeh et al. \(2020\)](#) tackles uncertainty estimation by incorporating Bayesian methods directly into the object detection process. This approach reduces uncertainty error metrics, particularly at the Non-Maximum Suppression (NMS) stage, but it still depends on a Bayesian framework and does not provide statistical guarantees.

While these methods offer valuable contributions to uncertainty estimation, they all come with limitations, especially regarding the need for modifications to the training process, computational complexity, and lack of statistical certificates. In contrast, Conformal Prediction (CP) offers a more promising alternative by providing performance guarantees without requiring such significant changes to the model training pipeline.

3.2. Conformal Prediction for Object Detection and Aerospace

Conformal Prediction (CP) is a flexible, distribution-free framework that enables the construction of statistically valid prediction sets, under the minimal assumption that data are exchangeable [Vovk et al. \(2005\)](#); [Angelopoulos et al. \(2024\)](#). In its standard form, CP requires a trained model and a separate calibration dataset to compute nonconformity scores—measures of how unusual a new prediction is compared to previously observed outcomes. The result is a prediction set that, with probability approximately $1 - \alpha$, contains the true label. This makes CP particularly attractive for safety-critical applications where quantifying predictive uncertainty is essential.

Extending CP to object detection poses unique challenges, primarily due to the need to conformalize structured outputs like bounding boxes. Recent works have explored ways to adapt CP by introducing pairing strategies between predicted and ground-truth boxes, typically based on IoU thresholds [de Grancey et al. \(2022\)](#); [Andéol et al. \(2023\)](#). Some approaches apply CP box-wise, computing per-coordinate or per-box nonconformity scores, while others propose more elaborate frameworks such as two-step conformal prediction to handle uncertainty in both localization and classification [Timans et al. \(2024\)](#). These methods aim to produce conformalized bounding boxes that maintain the desired coverage while remaining practically tight and adaptive. Robust variants of conformal prediction have also been developed to maintain valid uncertainty guarantees under adversarial perturbations or distribution shifts [Massena et al. \(2025\)](#).

In the aerospace domain, CP has been applied across multiple modalities, including surrogate modeling [Ducoffe et al. \(2020\)](#), satellite image analysis [Copley et al. \(2024\)](#), and air traffic control systems [Ernez et al. \(2023\)](#).

4. Single Object Detection for Visual Based Landing

The Vision Landing System (VLS) pipeline consists of three core stages, each critical for the accurate detection and localization of an aircraft during landing. The full pipeline is captured by Figure 3. It is considered and has been revisited by various companies and researchers in the aeronautic field ([Balduzzi et al. \(2021\)](#), [Dai et al. \(2024\)](#)):

- **Stage 1 – Object Detection (OD)**

The system receives an input image and identifies the runway within it. This detected region of interest is then cropped to enhance precision in subsequent stages. This step involves training an object detection model to locate and identify runways

- **Stage 2 – Feature Regression**

From the cropped image, the system predicts runway features—such as its four corners. These features are used to define the runway’s geometry and establish a mapping between 2D image coordinates and their corresponding 3D positions in the real world.

- **Stage 3 – Pose Estimation**

Using the extracted features, the system estimates the aircraft’s pose relative to the runway. This is accomplished using pose estimation techniques, providing real-time aircraft localization during the landing phase.

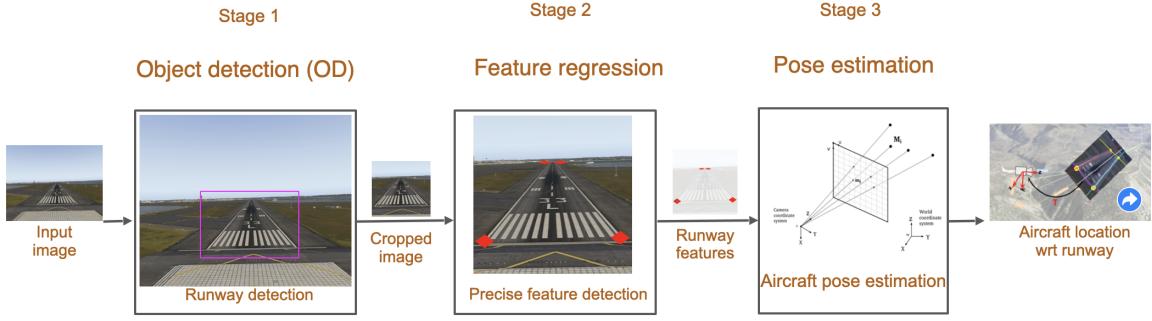


Figure 3: Overview of the Visual Landing System (VLS) pipeline. The system processes an input image through three stages: (1) Object Detection to identify and crop the runway region, (2) Feature Regression to extract key geometric features such as corners, and (3) Pose Estimation to compute the aircraft’s position relative to the runway using the extracted features. Each stage builds upon the previous one, with accurate detection being critical to the reliability of the final pose estimation.

Each stage plays a crucial role in ensuring accurate aircraft localization. In particular, the stage of **Runway Detection (Stage 1)** forms the foundation of the entire pipeline. Errors at this stage can propagate through the system, degrading both feature extraction and pose estimation accuracy. Therefore, the precision of the initial detection has a direct and significant impact on overall system performance. To ensure reliable corner detection, the cropped image must contain the **full extent** of the runway and be **small enough** to avoid inaccuracies introduced by interpolation. Maintaining this balance is essential for preserving an accurate pose estimation.

4.1. On Robust Detection Metrics: IoU vs. IoA

Given the importance of accurately identifying the runway region in Stage 1, it is crucial to evaluate object detection performance using metrics that reflect both spatial precision and complete coverage. Our base predictor, which we aim to conformalize, is trained to maximize mean Average Precision (mAP), a metric driven primarily by IoU. While IoU is the conventional choice for object detection evaluation, it does not fully align with the dual objectives of our system illustrated in Figure 4.

1. The predicted bounding box should *fully enclose* the ground truth box to ensure reliable feature regression and pose estimation. This is particularly critical, as pose estimation becomes unreliable with fewer than three visible runway corners.
2. The predicted bounding box should also be *accurately aligned* with the ground truth to avoid interpolation error when estimating key point features.

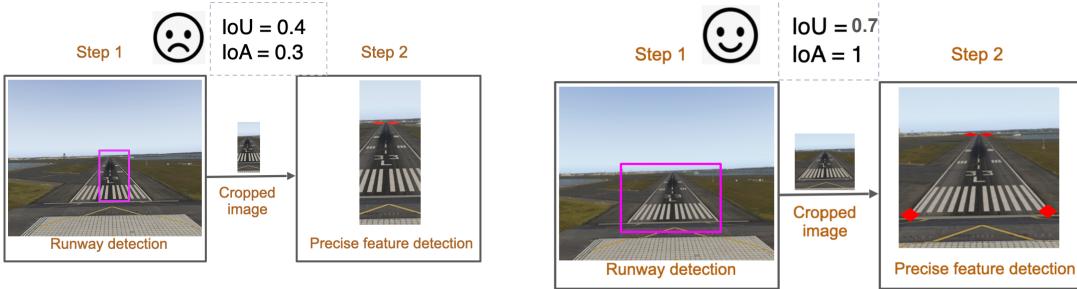


Figure 4: Pipeline examples illustrating different IoU and IoA outcomes.

IoA has been previously employed in the conformalized object detection literature (e.g., de Grancey et al. (2022)) due to its *monotonicity*. This property implies that increasing the size of the predicted box to better enclose the ground truth results necessarily in a higher IoA score. In contrast, IoU may decrease in such scenarios if the added area extends beyond the ground truth, making IoA more compatible with Conformal Prediction (CP) frameworks that require such predictable behavior. We define a detection as *robust* when both spatial accuracy (IoU) and full coverage (IoA) are satisfied:

$$\text{Robust Detection} \iff \text{IoU}(\mathbf{b}_{\text{pred}}, \mathbf{b}_{\text{gt}}) \geq t \quad \text{and} \quad \text{IoA}(\mathbf{b}_{\text{pred}}, \mathbf{b}_{\text{gt}}) = 1,$$

where \mathbf{b}_{pred} and \mathbf{b}_{gt} denote the associated predicted and ground truth bounding boxes, respectively. Our conformal predictor aims to guarantee the $\text{IoA} = 1$ condition for robust detection, while relying on the base predictor to maintain high IoU. However, introducing such a guarantee inevitably raises the question of how much the IoA constraint degrades the IoU performance. To summarize this relationship and motivate our evaluation, Table 1 provides an overview of how standard YOLO and our conformalized YOLO (C-YOLO) models satisfy the conditions required for a robust detection.

Table 1: Summary of robustness criteria for YOLO and C-YOLO models.

	$\text{IoU} > t$	$\text{IoA} = 1$	Robust?
YOLO	✓	✗	✗
C-YOLO	?	✓	?

4.2. Conformal Average Precision (C-AP)

In Conformal Single Object Detection, where we focus on detecting a single class—specifically the runway—we introduce Conformal Average Precision (C-AP). Our method is straightforwardly applicable to multiple object detection tasks, where the C-mAP metric can evaluate performance across multiple classes. Our approach proceeds in two steps:

1. We apply a Conformal Prediction (CP) framework to ensure that the IoA equals 1 for the majority of true positive predictions.

2. We introduce a new metric, *Conformal Average Precision (C-AP)*, to quantify the trade-off between constraints (1) and (2) using a modified version of Average Precision (AP).

First we adopt a conformal prediction (CP) framework that leverages the discrepancy between predicted and ground-truth bounding box coordinates. Specifically, we follow the method proposed by [de Grancey et al. \(2022\)](#), where the nonconformity score is defined based on coordinate-wise differences between predicted and groundtruth bounding boxes.

Let each ground-truth bounding box be indexed by $k = 1, \dots, N_{gt}$, regardless of the image it appears in. We denote the ground-truth coordinates of the k -th box as $\mathbf{b}_k^{gt} = (x_{\min}^k, y_{\min}^k, x_{\max}^k, y_{\max}^k)$, and its predicted counterpart as $\hat{\mathbf{b}}_k = (\hat{x}_{\min}^k, \hat{y}_{\min}^k, \hat{x}_{\max}^k, \hat{y}_{\max}^k)$. The **additive** nonconformity measure \mathbf{r}_k is then defined as the vector:

$$\mathbf{r}_k^a = \left(\hat{x}_{\min}^k - x_{\min}^k, \hat{y}_{\min}^k - y_{\min}^k, \hat{x}_{\max}^k - \hat{x}_{\min}^k, \hat{y}_{\max}^k - \hat{y}_{\min}^k \right).$$

while the **multiplicative** nonconformity measure \mathbf{r}_k is defined using the width and height of the predicted bounding box $\hat{w} = \hat{x}_{\max} - \hat{x}_{\min}$, $\hat{h} = \hat{y}_{\max} - \hat{y}_{\min}$

$$\mathbf{r}_k^m = \left(\frac{\hat{x}_{\min}^k - x_{\min}^k}{\hat{w}}, \frac{\hat{y}_{\min}^k - y_{\min}^k}{\hat{h}}, \frac{x_{\max}^k - \hat{x}_{\max}^k}{\hat{w}}, \frac{y_{\max}^k - \hat{y}_{\max}^k}{\hat{h}} \right).$$

Similarly as previous approaches, we use the Hungarian matching algorithm during the conformalization of predicted boxes.

Once these pairings are established, we compute nonconformity scores for conformal prediction. Since our bounding box predictions are represented as 4-dimensional vectors (one value for each side of the box), we obtain four separate distributions of nonconformity scores—unlike the scalar scores used in standard conformal prediction. To maintain valid statistical coverage guarantees across all dimensions, we apply a Bonferroni correction, as originally proposed in [de Grancey et al. \(2022\)](#). That is, we build quantiles for each of the 4 non-conformity component, at level $\frac{\alpha}{4}$, resulting in

$$\hat{q}_j = q_{\lceil(1-\frac{\alpha}{4})(n+1)\rceil/n}(\{\mathbf{r}_k(j) : k \in \{1, \dots, n\}\}),$$

where $q_{\lceil(1-\frac{\alpha}{4})(n+1)\rceil/n}$ is the *corrected* quantile function at level $\frac{\alpha}{4}$, and $r_k(j)$ as well as \hat{q}_j are respectively the j -th component of the vector r_k and \hat{q} .

The resulting quantiles from each score distribution are then used to adjust predictions at inference by modifying the sides of each predicted box $\hat{\mathbf{b}}_k$. In the additive case, we construct the conformal bounding box $\hat{\mathbf{b}}_k^{\text{conf}}$ by adding or subtracting the corresponding quantile values to the sides of $\hat{\mathbf{b}}_k$, thus expanding the box by fixed pixel margins. In the multiplicative case, $\hat{\mathbf{b}}_k^{\text{conf}}$ is obtained by scaling these quantile values by the predicted box width and height, so that the margins grow proportionally with the size of $\hat{\mathbf{b}}_k$. This ensures that the conformal prediction intervals adapt to the object size, as proposed in [de Grancey et al. \(2022\)](#). In the **additive case**, we apply fixed pixel offsets:

$$\hat{\mathbf{b}}_k^{\text{conf}} = \left(\hat{x}_{\min}^k - \hat{q}_1, \hat{y}_{\min}^k - \hat{q}_2, \hat{x}_{\max}^k + \hat{q}_3, \hat{y}_{\max}^k + \hat{q}_4 \right).$$

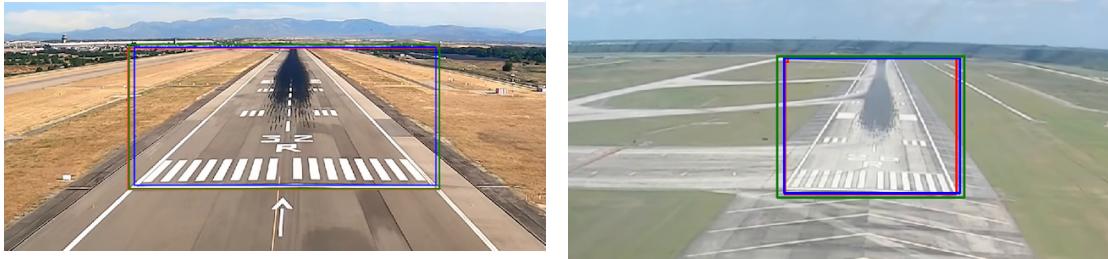


Figure 5: Conformalized inputs for YOLOv5 (left) and YOLOv6 (right), shown in green, generated using the additive method.

And in the **multiplicative case**, the offsets are scaled by the predicted box dimensions:

$$\hat{\mathbf{b}}_k^{\text{conf}} = \left(\hat{x}_{\min}^k - \hat{q}_1 \hat{w}, \hat{y}_{\min}^k - \hat{q}_2 \hat{h}, \hat{x}_{\max}^k + \hat{q}_3 \hat{w}, \hat{y}_{\max}^k + \hat{q}_4 \hat{h} \right).$$



Figure 6: Conformalized inputs for YOLOv5 (left) and YOLOv6 (right), shown in green, generated using the multiplicative method.

The resulting quantiles from each score distribution are then used to adjust predictions at inference. For a predicted bounding box $\hat{\mathbf{b}}_k$, we construct the *conformal* bounding box $\hat{\mathbf{b}}_k^{\text{conf}}$ using the quantiles $\hat{\mathbf{q}}$ following additive and multiplicative corrections of [de Grancey et al. \(2022\)](#).

The conformal bounding box is theoretically guaranteed to contain the true bounding box with a user-specified probability at least $1 - \alpha$. More formally, we have

$$\mathbb{P}\left(\mathbf{b}_{n+1}^{gt} \subseteq \hat{\mathbf{b}}_{n+1}^{\text{conf}}\right) \geq 1 - \alpha,$$

where the $b \subseteq b'$ corresponds to the box inclusion (i.e., holding true if $x_{\min} \geq x'_{\min}$, $y_{\min} \geq y'_{\min}$, $x_{\max} \leq x'_{\max}$, $y_{\max} \leq y'_{\max}$).

In the traditional Average Precision (AP) framework, a predicted box $\hat{\mathbf{b}}_k$ is considered a true positive if it satisfies $\text{IoU}(\hat{\mathbf{b}}_k, \mathbf{b}_k^{gt}) \geq t$. In our conformal AP (C-AP) formulation, we strengthen this condition, as illustrated in Figure 7:

$$\text{C-AP Match} \iff \text{IoU}(\hat{\mathbf{b}}_k, \mathbf{b}_k^{gt}) \geq t \quad \text{and} \quad \text{IoA}(\hat{\mathbf{b}}_k, \mathbf{b}_k^{gt}) = 1.$$

This stricter requirement ensures that a prediction is only counted as a true positive if it not only sufficiently overlaps the ground truth but also fully contains it—better aligning detection evaluation with the downstream requirements of accurate pose estimation.

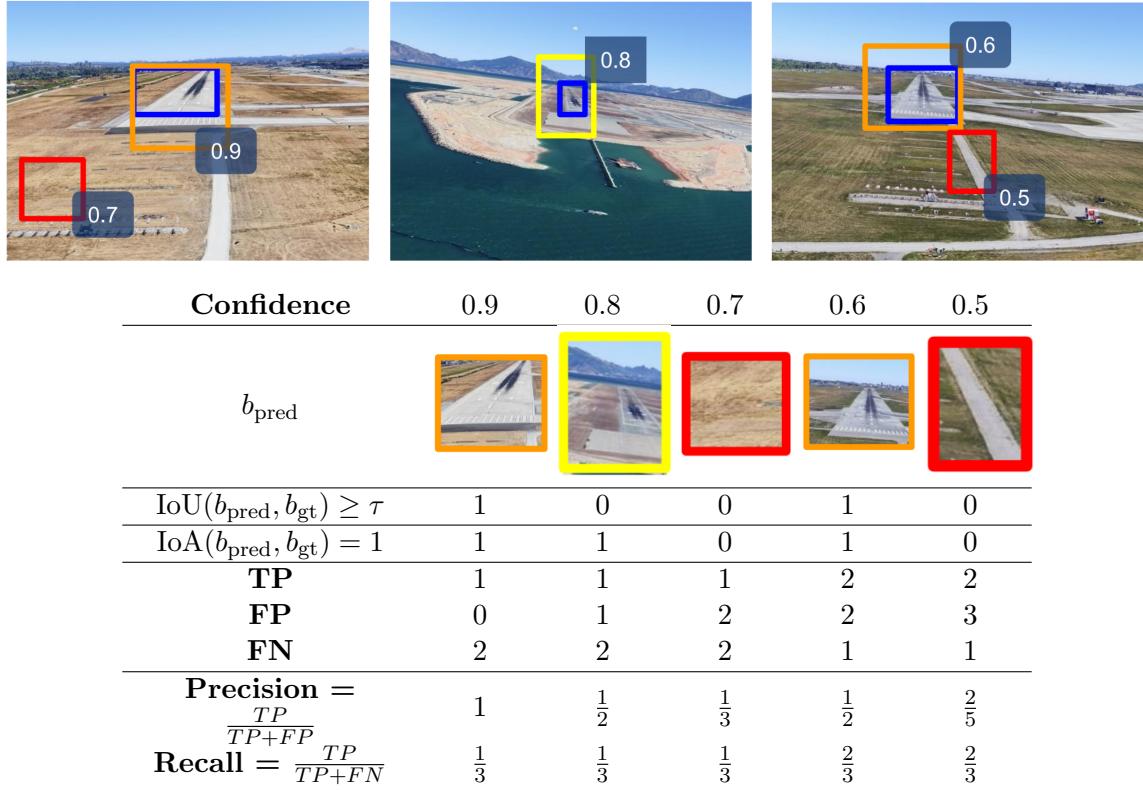


Figure 7: Computation of **C-(m)AP** step-by-step on an example from the LARD dataset. The top image shows the visual setup, while the table illustrates how predictions at different confidence levels affect precision and recall.

5. Experiments

Our experiments seek to demonstrate the utility of conformal prediction approaches to quantify localization uncertainty which exist in object detection models.

Dataset: the LARD dataset [Ducoffe et al. \(2023\)](#) comprises high-quality aerial images of runway during approach and landing phases. We split the dataset into train, validation and test. The base predictor used to compute the mAP is trained on the training set and evaluated on the validation set. We also kept an independent set, which we call the test set. This set is used exclusively for our conformal prediction experiments. We split it into an 80/20 partition, where 80% is used for the calibration process and 20% for evaluating the conformal predictors by calculating metrics such as Stretch, Coverage, and C-mAP. The mAP of the models are evaluated on the validation set while the experiments on conformal prediction are done on the the remaining 20% of the test set. The number of image per split is described in Table 2. The LARD dataset contains both synthetic images and real images but the real images are only kept in the test set. The current version of the dataset contains only one ground truth runway annotation per image. However, our method is already designed to generalize to multiple ground truth instances if they become available in future versions.

Set	Type	Images
Train	Synthetic	11,546
Validation	Synthetic	2,886
Test	Real+Synth	2,315

Table 2: Dataset composition by set, type, and number of images.

5.1. Models Description and Evaluation

In this section, we describe the object detection models employed for the runway detection task and evaluate their performance on aerial imagery. We benchmark two state-of-the-art, pre-trained models from the YOLO (You Only Look Once) family: YOLOv5-small [Jocher et al. \(2020\)](#) and YOLOv6-small [Li et al. \(2022\)](#), chosen for their balance between detection accuracy and computational efficiency—an essential requirement for embedded vision-based landing systems.

5.1.1. MODEL CONFIGURATIONS AND TRAINING

Both models were initialized from pre-trained weights available through their respective official repositories and fine-tuned on our runway detection dataset using standard training protocols. YOLOv5-small was trained for 100 epochs with a batch size of 16 and an input resolution of 640×640 pixels. This configuration aligns with common practices in the YOLO community and was found sufficient for convergence. YOLOv6-small was trained for 97 epochs, using a batch size of 32 and the same input resolution of 640×640 . Training was conducted using the official YOLOv6 repository and default optimization settings.

Both training procedures incorporated extensive data augmentation, which is critical for generalization on aerial imagery with varying lighting, weather, and perspectives, using the default configuration.

5.1.2. MODEL EVALUATION

The models were evaluated using standard object detection metrics: mean Average Precision at IoU 0.5 (mAP@0.5) and mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95). These metrics offer insights into both loose and strict detection accuracies. Additionally, we report GFLOPs (Giga Floating Point Operations) as a proxy for computational efficiency—an important constraint for real-time, onboard inference.

As shown in Table 3, both models achieve near-perfect performance on the runway detection dataset. YOLOv5 achieves slightly higher precision, especially at tighter IoU thresholds, while YOLOv6 presents a more computationally demanding architecture, as indicated by its GFLOPs. These results provide a strong baseline for applying conformal prediction methods, where bounding box uncertainty will be layered atop already accurate detections. Figure 8 displays a scatter plot comparing the performance of YOLOv5 and YOLOv6. The x-axis represents the IoU between YOLOv5’s predictions and the ground-truth boxes, while the y-axis shows the IoU for YOLOv6. The distribution of points suggests no clear correlation between the IoU values of the two models. This indicates that the models behave differently across images, often producing divergent predictions for the same inputs.

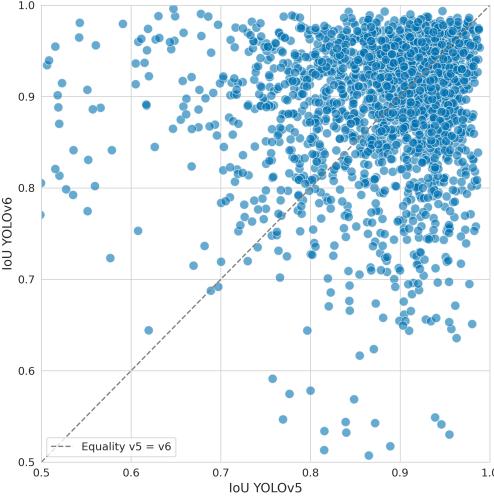


Figure 8: Scatter plot comparing the IoU of YOLOv5 (x-axis) and YOLOv6 (y-axis) with respect to the ground-truth boxes. Each point represents a single image. The lack of a clear correlation suggests that the two models produce different predictions on the same inputs.

Table 3: Comparison of pre-trained object detection models on runway detection task evaluated on the validation dataset.

Metric	YOLOv5 (pre-trained)	YOLOv6 (pre-trained)
mAP@0.5	0.995	0.9901
mAP@0.5:0.95	0.9712	0.9413
GFLOPs	15.8	45.3

5.2. Conformal Prediction for Object Localization

We evaluate the effect of conformalization on object detection models by analyzing the coverage and bounding box area after applying conformal prediction with error level $\alpha = 0.3$. We use the PUNCC library Mendil et al. (2023). We define respectively by c-YOLOv5- a, m and c-YOLOv6- a, m the conformalized versions of the original YOLOv5 and YOLOv6 models, with a denoting the additive penalty and m the multiplicative penalty used to enlarge predicted bounding boxes. Table 6 reports the proportion of predicted boxes that cover their associated ground truth (coverage), along with the average area and variance of the conformalized predicted boxes.

Our results show that the empirical coverage often exceeds the target level $(1 - \alpha)$. This conservativeness is likely a result of the Bonferroni correction, which inflates the quantiles to ensure joint coverage over multiple dimensions. This however, results in slightly larger box margins. Consequently, depending on the baseline accuracy of the detection model, enforcing stricter coverage (lower α) or using more conservative nonconformity scores will generally produce larger bounding boxes to maintain the desired guarantees. Among the evaluated models, c-YOLOv5-a achieves the highest coverage at 77.06%, followed by c-

YOLOv5-m (75.88%), c-YOLOv6-a (75.73%), and c-YOLOv6-m (73.93%). This indicates that the additive penalty tends to provide slightly better coverage than the multiplicative one, particularly for YOLOv5.

Our next study will compare the Conformal Average precision of all models. We evaluate the performances of standard YOLO models (YOLOv5 and YOLOv6) and their conformalized variants (c-YOLOv5 with additive penalty, c-YOLOv5 with multiplicative penalty, c-YOLOv6 with additive penalty, and c-YOLOv6 with multiplicative penalty) across three metrics: standard mean Average Precision (mAP), Conformal mean Average Precision (C-mAP, naturally extended from C-mAP), and C-mAP@50@80:100, a variant measuring performance with IoU threshold set to 0.5 and across varying IoA thresholds from 0.8 to 1.0. Metrics are computed on the Test Set.

The results, reported in Table 4, reveal clear trends:

Model	mAP	C-mAP	C-mAP@50@80:100
YOLOv5	96.88	0.77	46.92
c-YOLOv5-a	92.67	56.86	80.73
c-YOLOv5-m	96.17	55.84	82.18
YOLOv6	98.13	1.31	51.94
c-YOLOv6-a	95.09	55.75	81.86
c-YOLOv6-m	96.71	52.71	81.93

Table 4: Evaluation of YOLOv5 and YOLOv6 and their conformalized version on the Global Test Dataset

First, standard YOLO models achieve very high mAP scores (96.88% for YOLOv5 and 98.13% for YOLOv6), confirming their excellent performance under traditional detection criteria based solely on IoU. However, their C-mAP scores are extremely low (0.77% and 1.31%, respectively), indicating that very few predictions fully contain the ground truth boxes as required by the stricter conformal matching condition. This gap highlights that conventional detectors, while spatially accurate in terms of overlap, often fail to completely encapsulate objects—a critical property for downstream tasks such as precise pose estimation.

In contrast, the conformalized models significantly improve C-mAP scores, achieving between 52.71% and 56.86%, while maintaining high standard mAP values (above 92% in all cases). This demonstrates that the conformalization process, whether via an additive or a multiplicative penalty, effectively adapts the models to produce predictions that not only overlap but also fully contain ground truths, with only a minor compromise on traditional detection performance. The intermediate metric, C-mAP@50@80:100, further supports this observation. While vanilla YOLO models achieve moderate scores (46.92% and 51.94%), their conformalized counterparts reach substantially higher results (approximately 81–82%). This suggests that conformalized models remain robust across a range of IoA thresholds, consistently maintaining high-quality detections even as the containment requirement becomes more stringent.

Finally, between the additive and multiplicative penalty variants, the multiplicative models generally achieve a slightly better balance, preserving higher standard mAP scores while maintaining competitive C-mAP and C-mAP@50@80:100 performances.

In summary, these results demonstrate the effectiveness of conformalization in adapting object detectors to stricter, application-driven evaluation criteria. Furthermore, they highlight the relevance of our proposed Conformal mAP (C-mAP) metric, which simultaneously encapsulates both precision (through IoU requirements) and full coverage (through IoA constraints), offering a more holistic and practical assessment of detection performance for downstream applications where spatial guarantees are crucial.

In order to quantify the impact of conformalization over our base predictor, we compute the margin and expansion of the conformalized boxes as:

$$\text{Margin}(d) : \quad \frac{1}{n} \sum_{i=1}^n \left| \hat{\mathbf{b}}_{i,d}^{\text{conf}} - \hat{\mathbf{b}}_{i,d} \right|,$$

where $\hat{\mathbf{b}}_{i,d}^{\text{conf}}$ and $\hat{\mathbf{b}}_{i,d}$ correspond respectively to the d -th index of the i -th conformalized and predicted box, and n the number of boxes.

The margins in Table 5 show how much the predicted boxes were expanded to form the conformalized boxes, with higher values indicating significant expansion and smaller values suggesting minimal adjustments. Balanced margins reflect symmetrical corrections. c-YOLOv5-a and c-YOLOv6-a have smaller margins, with c-YOLOv6-a exhibiting the least expansion across all directions. This suggests high model confidence and minimal adjustments, indicating better prediction accuracy. In contrast, c-YOLOv5-m requires larger adjustments, especially in the left (18.17) and right (12.04) directions, implying lower model confidence and more substantial corrections. The Total Average column in Table 5 provides an overall measure of expansion across all directions. A lower value indicates more precise predictions, while a higher value suggests more adjustments. c-YOLOv6-a has the lowest total average margin (7.03), reflecting better accuracy, while c-YOLOv5-m has the highest total average margin (12.18), indicating the model requires larger corrections.

Model	Left	Top	Right	Bottom	Total Average
c-YOLOv5-a	11.76	5.79	8.93	8.54	8.76
c-YOLOv5-m	18.17	7.85	12.04	10.65	12.18
c-YOLOv6-a	10.39	3.47	7.76	6.52	7.03
c-YOLOv6-m	15.58	4.92	11.75	7.32	9.89

Table 5: Average predictive margins (in pixels) per side for each conformalized YOLO model.

Furthermore, we calculate the Stretch metric (e.g., Andéol et al. (2023)), which quantifies the expansion ratio of the conformalized box area relative to the predicted box area. This metric provides insight into the confidence of the model’s predictions, helping to determine whether the model is making small, precise adjustments or compensating for under-covering predictions with larger corrections. The formula is as follows: $\text{Stretch} = \frac{1}{n} \sum_{i=1}^n \sqrt{\frac{\text{Area}(C_i)}{\text{Area}(P_i)}}$.

Where $\text{Area}(C_i)$ is the area of the conformalized box C_i , and $\text{Area}(P_i)$ is the area of the predicted box P_i .

A Stretch value near 1 indicates minimal expansion of the predicted box, suggesting high model confidence with only slight adjustments needed. In contrast, a significantly higher stretch value points to a larger expansion of the predicted box, typically indicating greater uncertainty in the prediction. This suggests that the model compensates for an under-covering prediction by expanding the box to ensure the entire object is captured, reflecting the model’s uncertainty about the object boundaries.

In the results (Table 6), c-YOLOv6-m shows the lowest Stretch value (1.11), indicating that the model’s predictions are the closest to the conformalized boxes, with minimal adjustment needed. c-YOLOv5-a, on the other hand, has a Stretch value of 1.16, indicating slightly more expansion, which suggests a bit more uncertainty or adjustments. Regarding Variance and the mean of the square root of box area, we observe that c-YOLOv6-a has the smallest variance (156.60) and mean area (188.76), suggesting that it exhibits more stable and consistent box areas across predictions. In contrast, c-YOLOv5-m shows a higher variance (177.45) and area (200.42), indicating that the predicted boxes are more spread out and less consistent, reflecting greater variability in the model’s predictions.

Model	Stretch	$\sqrt{\text{Box Area}}$	Coverage
c-YOLOv5-a	1.16	194.92 ± 156.74	77.06
c-YOLOv5-m	1.13	200.42 ± 177.45	75.88
c-YOLOv6-a	1.13	188.76 ± 156.60	75.73
c-YOLOv6-m	1.11	193.05 ± 173.47	73.93

Table 6: Evaluation of YOLOv5 and YOLOv6 on the test dataset with different conformalized box penalty, with associated coverage, squared root of box area and stretch for $\alpha = 0.3$.

To further understand the behavior of the models, we now turn to the visual analysis of the IoU and Intersection over Area (IoA) distributions. Figures 10 and 11 provide deeper insights into how the conformalized YOLO models compare to the standard YOLO models, specifically in terms of their precision, coverage, and the impact of conformalization on false positives and false negatives.

Figure 10 shows the relationship between IoU and IoA for conformalized YOLO models. It highlights that the majority of conformalized predicted boxes, with more than 70% (according to a user-defined ratio $\alpha = 0.3$), saturate at $\text{IoA} = 1$. This means that, for most predictions, the conformalized models meet the full coverage criterion ($\text{IoA} = 1$), which is a key feature of the conformalization process. Consequently, at most 30% of the predicted boxes are likely to result in false negatives, as they fail to fully cover the ground truth. This behavior explains why the C-mAP scores for the conformalized models are consistently lower than 70%, despite their improvement over standard YOLO models. To achieve higher C-mAP scores, users can decrease the value of α , which helps reduce the number of false positives as measured in C-mAP.

Although the nominal risk level is set at $\alpha = 0.3$, achieving a slightly higher empirical coverage, such as 77.06%, does not necessarily imply a problematic increase in box size. These higher coverage values are a consequence of conservative adjustments, including the

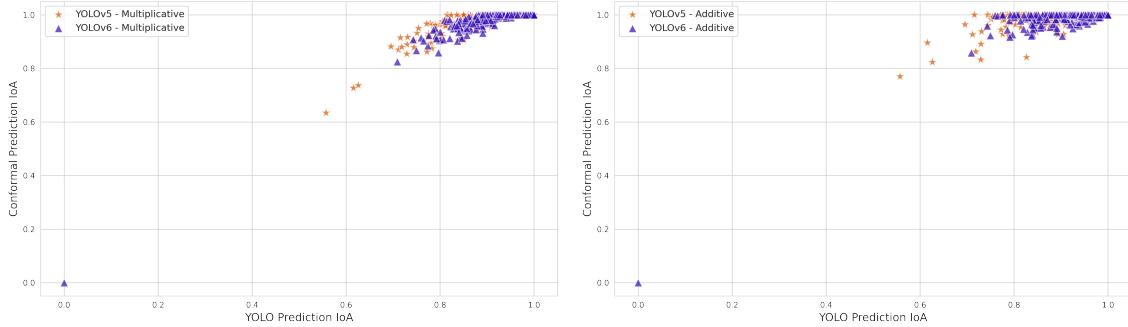


Figure 9: Scatter plots comparing the Intersection over Area (IoA) of YOLOv5 and YOLOv6 predictions (x-axis) with their conformalized versions. Left: multiplicative penalty (boxes expanded by a percentage). Right: additive penalty (boxes enlarged by a fixed number of pixels).

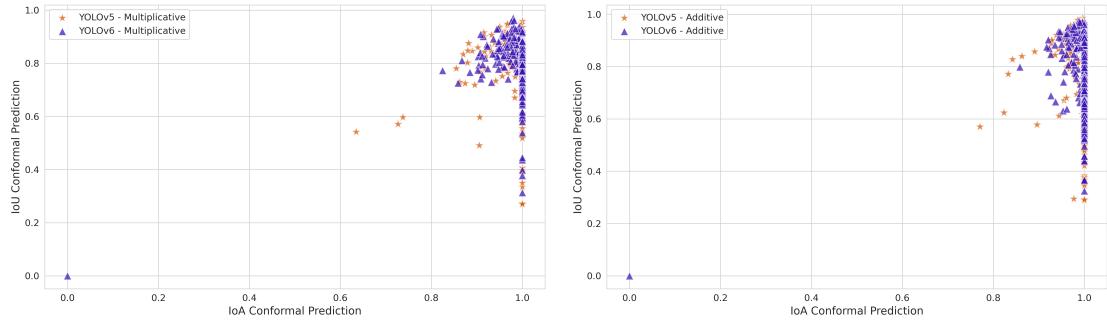


Figure 10: Scatter plots comparing the IoU of YOLO’S predictions (x-axis) with the IoA of their conformalized versions. Left: multiplicative penalty. Right: additive penalty.

Bonferroni correction applied in high-dimensional settings, which slightly enlarge the prediction regions to ensure reliable coverage guarantees. While conformalization can lead to larger bounding boxes, our results show that the IoU thresholds remain well controlled, as evidenced by the maintained high C-mAP scores. Therefore, even with some increase in box size, the conformalized models still produce sufficiently precise cropping of the images, ensuring good localization of key runway features without introducing interpolation or misalignment issues.

Figure 11 illustrates a slight degradation of IoU in the conformalized versions of YOLO, both for the additive and multiplicative penalty variants. Some of the conformalized predicted boxes have IoU values that drop below the threshold of 0.5 after conformalization, leading to an increase in false positives. This degradation results in a reduction in both mAP and C-mAP scores. Overall, c-YOLOv5-a provides the best balance between high coverage, controlled bounding box and C-mAP.

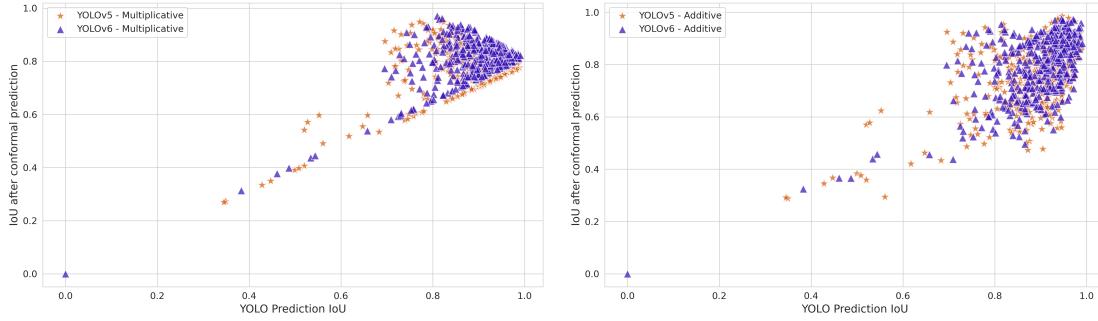


Figure 11: Scatter plots comparing the IoU of YOLO'S (x-axis) with their conformalized versions. Left: multiplicative penalty. Right: additive penalty.

6. Conclusion

Conformal Prediction (CP) offers a compelling alternative for uncertainty quantification. Unlike Bayesian or ensemble methods, CP requires no changes to the training process and provides formal, distribution-free guarantees. It can be applied to existing models, making it a practical and theoretically sound choice—especially in safety-critical applications where robustness and traceability are essential.

Conformal prediction delivers reliable uncertainty estimates with minimal overhead. However, it is not without limitations: it can produce overly conservative predictions—especially with corrections like Bonferroni adjustments—and its guarantees rely on calibration distributions matching test conditions. In multi-object detection, naive application can lead to artifacts like inflated bounding boxes or false positives. In practice, hybrid approaches—such as applying conformalization to the outputs of Bayesian models or deep ensembles—can combine the strengths of both paradigms, offering rich uncertainty estimates alongside formal, distribution-free coverage guarantees.

Finally, our current evaluation is limited by the LARD dataset, which contains only a single object per image. This restricts the analysis of the behavior of CP in multi-object scenes, where independent conformalization may lead to overlapping predictions and distort performance metrics such as mAP. Future work should investigate the impact of CP in more complex scenarios to fully assess its potential for dense detection tasks.

Acknowledgments

Our work has benefitted from the AI Interdisciplinary Institute ANITI. ANITI is funded by the France 2030 program under the Grant agreement n°ANR-23-IAACL-0002. The authors thank all the people and industrial partners involved in the DEEL project. This work has benefited from the support of the DEEL project,² with fundings from the Agence Nationale de la Recherche, and which is part of the ANITI AI cluster.

2. <https://www.deel.ai/>

References

- Léo Andéol, Thomas Fel, Florence De Grancey, and Luca Mossina. Confident object detection via conformal prediction and conformal risk control: an application to railway signaling. In *Conformal and Probabilistic Prediction with Applications*, pages 36–55. PMLR, 2023.
- Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *The Twelfth International Conference on Learning Representations*, 2024.
- Giovanni Balduzzi, Martino Ferrari Bravo, Anna Chernova, Calin Cruceru, Luuk van Dijk, Peter de Lange, Juan Jerez, Nathanaël Koehler, Mathias Koerner, Corentin Perret-Gentil, et al. Neural network based runway landing guidance for general aviation autoland. Technical report, United States. Department of Transportation. Federal Aviation Administration . . . , 2021.
- Puja Bharati and Ankita Pramanik. Deep learning techniques—r-cnn to mask r-cnn: a survey. *Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019*, pages 657–668, 2020.
- Vicky Copley, Greg Finlay, and Ben Hiett. The uncertain object: Application of conformal prediction to aerial and satellite images. In *The 13th Symposium on Conformal and Probabilistic Prediction with Applications*, pages 73–89. PMLR, 2024.
- Wei Dai, Zhengjun Zhai, Dezhong Wang, Zhaozi Zu, Siyuan Shen, Xinlei Lv, Sheng Lu, and Lei Wang. Yomo-runwaynet: A lightweight fixed-wing aircraft runway detection algorithm combining yolo and mobilerunwaynet. *Drones*, 8(7):330, 2024.
- Florence de Grancey, Jean-Luc Adam, Lucian Alecu, Sébastien Gerchinovitz, Franck Ma-malet, and David Vigouroux. Object detection with probabilistic guarantees. In *Fifth International Workshop on Artificial Intelligence Safety Engineering (WAISE 2022)*, 2022.
- Mélanie Ducoffe, Sébastien Gerchinovitz, and Jayant Sen Gupta. A high probability safety guarantee for shifted neural network surrogates. In *SafeAI@ AAAI*, pages 74–82, 2020.
- Mélanie Ducoffe, Maxime Carrere, Léo Féliers, Adrien Gauffriau, Vincent Mussot, Claire Pagetti, and Thierry Sammour. Lard-landing approach runway detection–dataset for vision based landing. *arXiv preprint arXiv:2304.09938*, 2023.
- Fares Ernez, Alexandre Arnold, Audrey Galametz, Catherine Kobus, and Nawal Ould-Amer. Applying the conformal prediction paradigm for the uncertainty quantification of an end-to-end automatic speech recognition model (wav2vec 2.0). In *Conformal and Probabilistic Prediction with Applications*, pages 16–35. PMLR, 2023.
- Ali Harakeh, Michael Smart, and Steven L Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 87–93. IEEE, 2020.

Glenn Jocher, Alex Stoken, Jirka Borovec, Liu Changyu, Adam Hogan, Laurentiu Diaconu, Jake Poznanski, Lijun Yu, Prashant Rai, Russ Ferriday, et al. ultralytics/yolov5: v3. 0. *Zenodo*, 2020.

Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

Zongyao Lyu, Nolan Gutierrez, Aditya Rajguru, and William J Beksi. Probabilistic object detection via deep ensembles. In *European Conference on Computer Vision*, pages 67–75. Springer, 2020.

Thomas Massena, Léo Andéol, Thibaut Boissin, Corentin Friedrich, Franck Mamalet, Mathieu Serrurier, and Sébastien Gerchinovitz. Efficient robust conformal prediction via lipschitz-bounded networks. 2025.

Mouhcine Mendil, Luca Mossina, and David Vigouroux. Punc: a python library for predictive uncertainty calibration and conformalization. In *Conformal and Probabilistic Prediction with Applications*, pages 582–601. PMLR, 2023.

Dimitry Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249. IEEE, 2018.

Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR’06)*, volume 3, pages 850–855. IEEE, 2006.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *The Second International Conference on Learning Representations*, 2014.

Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine learning and knowledge extraction*, 5(4):1680–1716, 2023.

Alexander Timans, Christoph-Nikolas Straehle, Kaspar Sakmann, and Eric Nalisnick. Adaptive bounding box uncertainties via two-step conformal prediction. In *European Conference on Computer Vision*, pages 363–398. Springer, 2024.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Simon Wenkel, Khaled Alhazmi, Tanel Liiv, Saud Alrshoud, and Martin Simon. Confidence score: The forgotten dimension of object detection performance evaluation. *Sensors*, 21(13):4350, 2021.

Sai Harsha Yelleni, Deepshikha Kumari, et al. Monte carlo dropblock for modeling uncertainty in object detection. *Pattern Recognition*, 146:110003, 2024.