AIRBUS

TOULOUSE INP N7 INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Enhancing Runway Detection Through Active Learning

Mehdi Benjid

MATHEMATICAL MODELING AND ARTIFICIAL INTELLIGENCE

Septembre 2023 - January 2024

School supervisor
Boris Djomgwe Teabe
boris.teabedjomgwe@enseeiht.fr

Company supervisor
Geoffrey Delhomme
geoffrey.g.delhomme@airbus.com

Acknowledgement

Throughout my academic journey, I am immensely grateful for the invaluable skills I have acquired, paving the way for a successful entry into the workforce, particularly through this internship. My sincere appreciation goes to INP-ENSEEIHT and INSA TOULOUSE, institutions that have not only provided a rich educational experience but have also dedicated efforts to the development of their students, offering the best possible foundation. I extend my gratitude to the exceptional teachers in the ModIA program, whose commitment to quality teaching and creation of a positive, enriching classroom environment has been instrumental.
A special acknowledgment is owed to the 1YISI team, with heartfelt thanks to my apprenticeship supervisor, David Soulié. His guidance introduced me to the dynamic working environment at Airbus, providing insights into the organization and the responsibilities of an engineer. His continuous support and encouragement have been pivotal in helping me leverage my skills effectively.
I extend my deepest thanks to the AI Plateau, particularly to my internship supervisors, Geoffrey Delhomme and Xavier Baril, for their warm welcome and invaluable guidance that contributed significantly to the success of my internship. My appreciation also goes to the entire AI Plateau team, under the leadership of Raphael Andre, for their kindness, creating an atmosphere that made me feel completely at ease and integrated into the team.

Table of Contents

1 Introduction **4**

2 Nomenclature **5**

3 Work Environment **6**
3.1 Airbus Organisation 6
3.1.1 Airbus : A Journey to Global Aerospace Power 6
3.1.2 Airbus Today : Leading Global Aerospace Innovation 6
3.1.3 Airbus' Vision : Sustainable Aviation and Innovation 7
3.2 Team Organisation 8
3.2.1 The Strategic Role Within Airbus 8
3.2.2 Main Project 8
3.2.3 Team Structure and Work Package Definitions 8
3.3 Development Workspace 9
3.3.1 Amazon EC2 for Scalability and Performance 9
3.3.2 Python Virtual Environment for a Clean and Isolated Development 10

4 Current Developments : Runway Detection using YOLOv8 **11**
4.1 Input Data 11
4.2 Runway Deep Learning-Based Detector 11
4.2.1 Introduction to YOLOv8 11
4.2.2 Precision Metrics in YOLOv8 : Understanding mAP and IoU . 12
4.2.3 Comparative Analysis : from Yolov5 to YOLOv8 13
4.2.4 Architectural Insights of YOLOv8 13
4.2.5 YOLOv8 Outputs : DIoU_NMS's Contribution to Precision . 16

5 Enhancing Runway Detection Using Active Learning **18**
5.1 Active Learning : An In-Depth Overview 18
5.1.1 Definition Active Learning 18
5.1.2 Exploring Active Learning Scenarios 18
5.2 YOLOv8's Features Extraction for Active Learning Strategies 20
5.3 Data Selection in Active Learning 20
5.3.1 Uncertainty Sampling Strategies 20
5.3.2 Diversity Sampling Strategy : The Coreset Approach 22

6 Practical Considerations and Strategies **23**
6.1 Integrating Active Learning with Deep Learning Models 23
6.2 Impact of Weight Re-Initialization 24

7 Experimental Results **25**
7.1 Performance of Active Learning Strategies 25
7.2 Weights Reinitialization 30

8 Conclusion and Discussion **34**

9 References **36**

Annex A : Metrics Definitions **38**

Annex B : Plots **40**

1 Introduction

In the dynamic and ever-evolving world of aviation, innovation and cutting-edge technology are imperative for enhancing safety, precision, and efficiency. One of the forefront projects in this domain is the Visual Landing System (VLS) initiated by Airbus. The VLS is a remarkable endeavor, which leverages the power of Artificial Intelligence (AI) to develop a sophisticated system capable of autonomously detecting the runway during an aircraft's landing phase. This novel approach aims to revolutionize the way aircraft navigate and touch down, ultimately bolstering flight safety and accuracy.

The primary objective of the VLS project is to harness the capabilities of deep learning-based models, a subset of AI, to enable aircrafts to autonomously identify and lock onto runways during the crucial phase of landing. This technology promises to be a game-changer, offering an advanced level of precision and reliability that was previously unimaginable. By implementing state-of-the-art AI models, Airbus seeks to reduce the dependence on manual intervention and human error, thereby augmenting safety standards and operational efficiency.

However, as with any ambitious project, the development of the VLS presents its own set of challenges. One of the critical issues to be addressed is the extensive data annotation and labeling required to train and fine-tune the deep learning models. To tackle this challenge, this report explores the integration of active learning as a strategic approach. Active learning is a data annotation methodology that selectively identifies and labels the most informative data points, significantly reducing the labor and cost associated with traditional manual annotation. This innovative approach is not only cost-effective but also aids in optimizing data storage, ensuring that the VLS remains an economically viable project without compromising on its performance.

This report delves into the intricate workings of a deep learning-based runway detector and offers insights into the application of active learning within the project.

2 Nomenclature

D Dataset
X Dataset Inputs
Y Dataset Output/Labels
K Set of Possible Classes
P Probability

3 Work Environment

3.1 Airbus Organisation

3.1.1 Airbus : A Journey to Global Aerospace Power

After World War II, the Western aeronautics sector experienced significant growth, with American companies like Lockheed and Boeing dominating the market. European enterprises struggled to compete, leading to a collaboration between France, the UK, and Germany in the late 1960s, resulting in the "Airbus A300" project in 1966.
France and Germany were given equal shares after the UK withdrew from the program in 1969. Airbus Industrie was founded by Sud-Aviation and Deutsche Airbus, with participation from CASA of Spain and British Aerospace of the UK.
In the 1980s and 1990s, the aviation market experienced exponential growth, leading to Airbus Industrie's rapid development. Key advancements included the A320 and A330/A340. To streamline management and reduce administrative costs, the consortium formed EADS (European Aeronautic Defence and Space Company), contributing to military, space, and helicopter construction through Eurocopter.
In 2014, EADS rebranded as Airbus Group, encompassing three main subsidiaries : Airbus Commercial Aircraft, headquartered in Blagnac, specializing in commercial aircraft ; Airbus Defense and Space, based in Munich, offering various products and services in the military and space sectors ; and Airbus Helicopters, headquartered in Marignane, focusing on helicopter production. This evolution reflects the journey of Airbus from its inception to its present-day status as a multifaceted leader in the global aerospace industry.

3.1.2 Airbus Today : Leading Global Aerospace Innovation

Today, Airbus is one of the global leaders in the aviation market, along with Boeing. In 2019, the group employed more than 130,000 employees worldwide, including 50,000 in France, 45,000 in Germany, 12,000 in Spain, and 11,000 in the United Kingdom. Its revenue amounts to 70.5 billion euros, equaling Boeing. In 2019, Airbus delivered more than 860 aircraft, surpassing Boeing and its 350 deliveries due to the 737MAX crisis. Airbus's fleet aims to be as comprehensive as possible to cover the entire market, just like its competitor Boeing (see figure 2).
Beyond its aircraft-related activities, Airbus is also involved in the military, space, and helicopter sectors. In fact, its subsidiary, Airbus Helicopters, accounts for nearly 10 % of the group's revenue and is the global leader in the market with a worldwide fleet estimated at 12,000 vehicles. Furthermore, the group is engaged in numerous space projects : it participated in the construction of the Pleiades satellites (CNES) in 2011, in the construction of Gaia (ESA) in 2013, and is slated for the construction of the JUICE satellite for Jupiter observation (ESA) in 2023. The group also holds

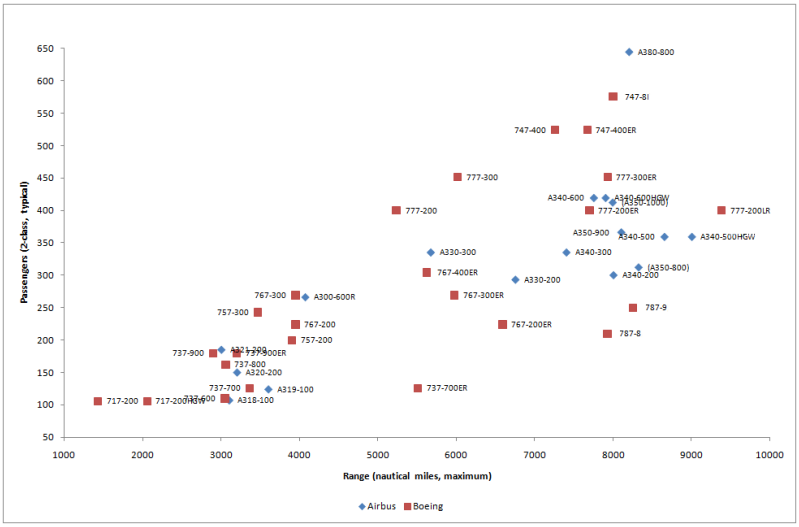

FIGURE 2 – Airbus and Boeing : Passengers vs Range

a 50% stake in ArianeGroup. Lastly, the group is also present in the military sector with the production of military aircraft such as the A400M or the Eurofighter, drones (EADS Harfang), and cybersecurity and communication services.

3.1.3 Airbus' Vision : Sustainable Aviation and Innovation

Airbus directs a significant portion of its resources toward innovation. Indeed, "At Airbus, we believe that the future of aerospace is autonomous, connected, and emission-free." Consequently, numerous projects for new zero-emission aircraft are emerging within the group. These encompass electric aircraft such as EcoPulse or E-Fan X, hydrogen-powered aircraft like the ZEROe projects (see figure 3), and disruptive aircraft such as the Blended-Wing Body, an innovative structure where the fuselage and wings have merged.

FIGURE 3 – Airbus ZEROe : Innovating for a Hydrogen-Powered Future

Airbus is also developing the Airbus Unmanned Traffic Management (UTM) project, which aims to enable autonomous operations of future aircraft (drones, air taxis) for seamless integration into our skies. Furthermore, Artificial Intelligence and Big Data are assuming increasingly prominent roles at Airbus, including their utilization to enhance future aircraft delivery estimates, thereby better satisfying customer needs.

3.2 Team Organisation

3.2.1 The Strategic Role Within Airbus

Over the past decade, artificial intelligence (AI) has become increasingly prevalent, yet Airbus only recently implemented its initial neural network on an aircraft, specifically the A380, just a couple of years ago. This introduction went relatively unnoticed, as it was initially deployed as a non-linear model. While various teams within Airbus have been exploring the integration of AI into their activities and decision-making processes, it wasn't until December 2021 that the AI Plateau was established. This initiative aims to bring together AI expertise with aeronautical knowledge to maximize the potential of AI in aviation and future Airbus projects. The AI Plateau collaborates with Acubed, Airbus' Silicon Valley innovation center, which was founded in 2015. As a crucial hub for innovation within the global aerospace leader, Acubed is dedicated to pioneering high-value, high-impact innovation projects, contributing to the future of the industry. The collaboration between the AI Plateau and Acubed signifies a strategic partnership aimed at harnessing and leveraging their combined AI knowledge for the benefit of Airbus and the aviation sector.

3.2.2 Main Project

The AI Plateau primarily focuses on developing a Visual Landing System (VLS) for various Airbus aircraft programs. This innovative system aims to assist pilots during the landing phase by identifying runway borders, determining the relative position of the aircraft, and predicting the trajectory. It is worth noting the substantial volume of data required to train a deep learning model effectively for these intricate tasks.

3.2.3 Team Structure and Work Package Definitions

The AI Plateau team is organized into distinct work packages (WP) to ensure an optimized and organized workflow :

- WP1 : Systems Requirements and Architecture (within A/C Architecture).
- WP2 : AI and Computer Vision Algorithms (design, development and testing).

- WP3 : Data Collection and Management, Synthetic Data Generation.
- WP4 : Target Hardware Platform and deployment tools.
- WP5 : V&V for vision (Massive testing and Integration tests).
- WP6 : Infrastructure and AI Development Environment.

The work presented in this report plays a crucial role between WP2, which focuses on Active Learning algorithms and model training, and WP3, which is centered around the management of data and the selection of data for further work. This positioning ensures a seamless integration of active learning strategies in the development and refinement of AI and computer vision algorithms, bridging the gap between algorithmic advancements and effective data utilization within the AI Plateau team's overall workflow.

3.3 Development Workspace

3.3.1 Amazon EC2 for Scalability and Performance

Training large deep learning models is computationally demanding, necessitating substantial computational resources. To address this challenge and attain scalability and high performance, we leverage a GPU-powered EC2 instance provided by Amazon Web Services.

Instance Configuration :

- GPU Model : four NVIDIA A10G GPUs
- Memory : each GPU features 24 GB of memory
- vCPUs : 48
- Storage type : Elastic Block Store (EBS)
- Storage capacity : 64 GiB
- Operating System : Amazon Linux 2
- AWS EC2 Instance Type : g5.12xlarge

This configuration provides a scalable balance between cost-effectiveness and efficiency in training large deep learning models, utilizing GPU acceleration to enhance performance and reduce training times.

3.3.2 Python Virtual Environment for a Clean and Isolated Development

In this project, we use a python virtual environment with conda (version 3.10.13). It allows to maintain a clean and isolated development environment, preventing version conflicts and dependencies from interfering with one another. With conda, we can easily create, activate, and deactivate virtual environments, making it a powerful tool for enhancing project reproducibility, dependency management, and overall development efficiency.

4 Current Developments : Runway Detection using YOLOv8

4.1 Input Data

The dataset consists of 16,244 images featuring airport runways, each with a resolution of 2048x2048 pixels. Each image is accompanied by a label that includes information for a bounding box : class_id (with '0' representing the single class of 'runway' in this study), x_center, y_center, width, and height. The dataset is split into two segments : 12,221 images are designated for training, and 4,023 images are reserved for validation purposes. Figure 4 illustrates an example of one such image, complete with its corresponding bounding box.

FIGURE 4 – Example of data

4.2 Runway Deep Learning-Based Detector

4.2.1 Introduction to YOLOv8

YOLOv8, as of early 2023, represents the latest advancement in the YOLO (You Only Look Once) series of models. Developed and maintained by the ultralytics team [1], YOLOv8 stands out for its ability to process images in a single forward pass, efficiently predicting all objects. This approach is a significant shift from traditional models that rely on region-based architectures, like RCNN, where object detection involves a two-step process : identifying regions of interest followed by classification.

YOLOv8's innovation lies in redefining object detection as a regression problem, focusing on predicting bounding box coordinates. This model is not only adept at excelling in classes it is pre-trained on, thanks to datasets like COCO[2] and ImageNet[3] , but also demonstrates remarkable adaptability to new classes. Its design makes it particularly suitable for embedded systems, showcasing the real strength of YOLOv8 in practical applications.
Renowned for its speed in both training and inference, YOLOv8 maintains high accuracy even with relatively compact model sizes. This efficiency makes it ideal for use on single GPUs, offering a powerful tool for both researchers and practitioners in the field of object detection.

4.2.2 Precision Metrics in YOLOv8 : Understanding mAP and IoU

In our analysis of the YOLO (You Only Look Once) object detection models, a pivotal metric to consider is the mean Average Precision (mAP). To grasp the essence of mAP, we first need to understand Precision in the realm of object detection. Precision is determined by the accuracy of the object detections, which is calculated using a metric called Intersection Over Union (IoU). As shown in figure 5, IoU measures the overlap between the predicted bounding box and the ground truth (actual) bounding box for an object. A higher IoU indicates a more accurate prediction.

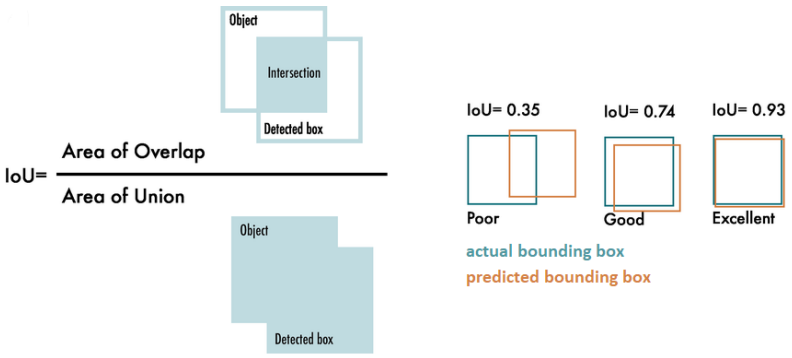

FIGURE 5 – Intersection over Union (IoU) [4]

For each object class, such as cars or pedestrians, Average Precision (AP) assesses the model's accuracy in predicting these classes across various IoU thresholds. The mAP is an aggregate measure, averaging the APs across all classes the model is trained to detect. Thus, a high mAP score denotes that the model reliably detects objects across a diverse set of classes with high precision. Comprehensive information on the computation of mAP and additional metrics utilized for assessing the performance of YOLOv8 can be found in Annex A (see page 38).

4.2.3 Comparative Analysis : from Yolov5 to YOLOv8

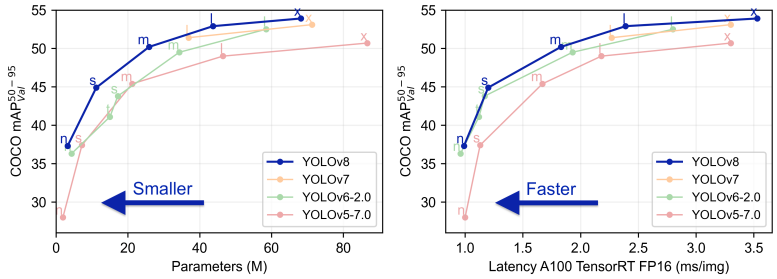

FIGURE 6 – Performance Analysis of YOLO Versions [5]

Figure 6 reveals that YOLOv8 boasts more parameters than its predecessors like YOLOv5 but fewer parameters than YOLOv6. Notably, it delivers approximately 33% higher mean Average Precision (mAP) for various model sizes, thereby achieving a consistently superior mAP performance across the spectrum. Furthermore, the second graph illustrates that YOLOv8 stands out for its faster inference times when compared to other YOLO models, reflecting its efficiency.

YOLOv8 presents a range of model sizes, each designated with specific suffixes, such as 'n' for nano, 's' for small, 'm' for medium, 'l' for large, and 'x' for extra-large, tailoring the models to accommodate a wide array of application requirements.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

FIGURE 7 – YOLOv8 Model Comparison : Size, Precision, Speed, and Computation Time [5]

In Figure 7, there is a comprehensive comparison of these model sizes with respect to their size, precision, speed, and computation time, providing a valuable reference for selecting the most suitable YOLOv8 variant for a given task.

4.2.4 Architectural Insights of YOLOv8

As YOLOv8 does not have an official paper as of now, direct insights into its research methodology and ablation studies are currently unavailable. Nevertheless,

we have conducted an analysis of the repository and available information to begin documenting the novel features introduced in YOLOv8. In this context, we offer a brief summary of the significant modeling updates, which will be followed by an examination of the model's performance evaluation, which serves as a testament to its capabilities.
To enhance understanding of the network's architecture, an informative image created by a GitHub user is provided (figure 8), offering a detailed visualization of the model's internal structure.

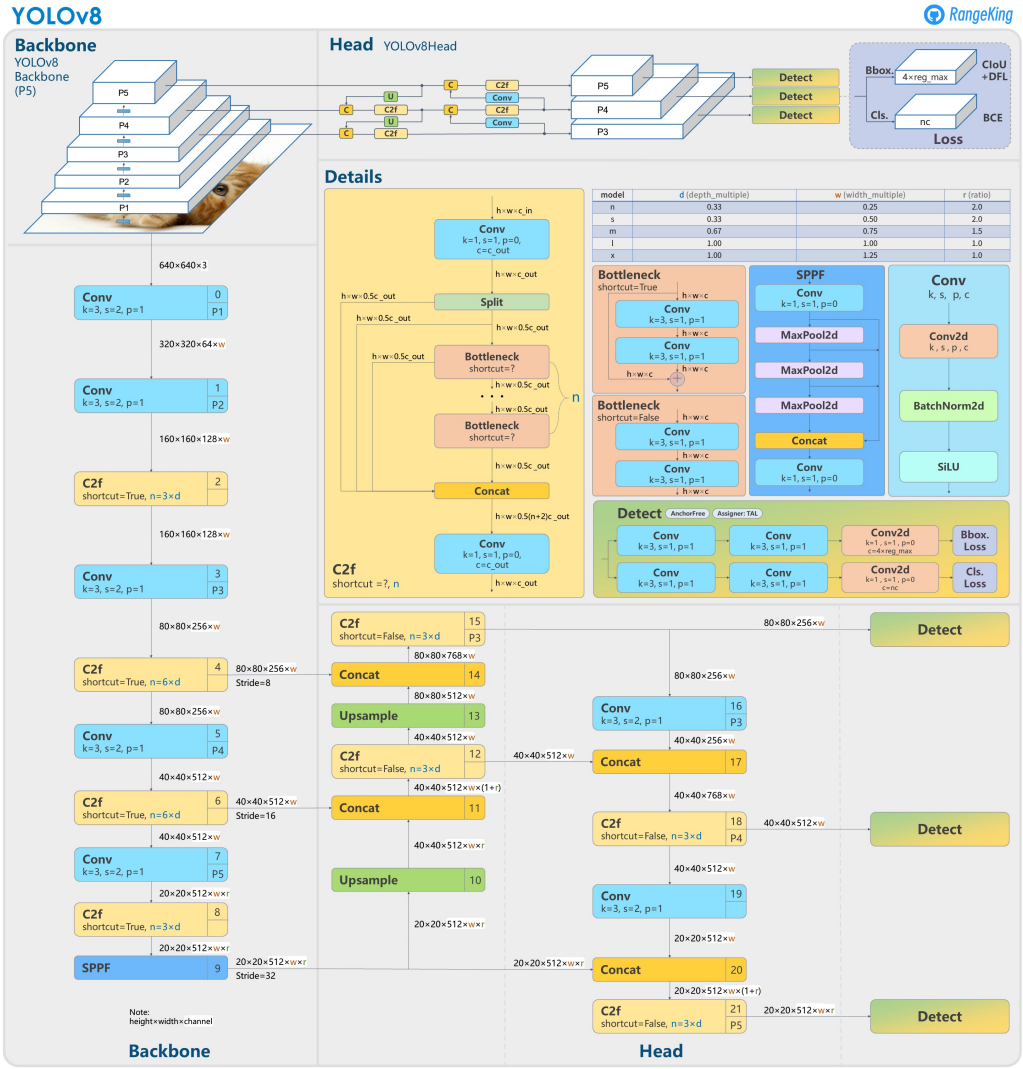

FIGURE 8 – YOLOv8 architecture

The YOLOv8 network is composed of three principal components :

- **The Backbone :** This section of the network is responsible for the initial

extraction of features from the input image. The backbone usually consists of a series of convolutional layers. Here, we can see that the input image is progressively downsampled through several stages (P1 to P5). This process is creating feature maps at different scales, which are then used in the detection process.

- **Head :** This part of the network is responsible for predicting bounding boxes, object classes, and objectness scores from the feature maps provided by the backbone. The head operates at different scales (P3 to P5) to detect objects of various sizes.

- **Detect :** this section of the network contains detection layers that predict bounding boxes, objectness and class probabilities . These are applied at different scales of the feature maps to ensure a wide range of object sizes can be detected.

YOLOv8 incorporates several layers, each contributing to the network's object detection capabilities :

- **Convolutional Layer (Conv) :** Applies filters to the input to create feature maps, with parameters specifying the kernel size, stride, and padding.

- **Cross-Channel Feature Pyramid (C2F) :** Merges features across channels and scales without extra parameters, facilitating multi-scale detection.

- **Bottleneck :** Reduces and then restores dimensions, often with residual connections to aid in training deep networks.

- **Concatenation (Concat) :** Merges feature maps from different layers to combine information at various scales.

- **Upsample :** Increases the resolution of feature maps to allow fine-grained detection at higher scales.

- **Spatial Pyramid Pooling - Faster (SPFF) :** Aggregates features at multiple scales to handle inputs of various sizes and maintain spatial details.

- **Batch Normalization (BatchNorm2d) :** Normalizes activations per feature map, improving training stability and speed.

- **Sigmoid Linear Unit (SiLU/Swish) :** Activation function combining sigmoid and linear characteristics, often providing better performance than traditional functions.

- **Max Pooling (MaxPool2d) :** Reduces the spatial dimensions of feature maps to decrease computational load and control overfitting.

These layers collectively process the input image, enabling the network to detect, classify, and pinpoint objects within the image.

4.2.5 YOLOv8 Outputs : DIoU_NMS's Contribution to Precision

The output tensor of the YOLOv8 model is a three-dimensional array with the shape float32[1, 84, 8400]. Let's break down each dimension :

- **Batch Size (1st Dimension) :** With a size of 1, the output corresponds to a single image. If there were multiple images processed together, this dimension would reflect the total number in the batch.

- **Feature Length (2nd Dimension) :** The length of 84 in this dimension is the concatenation of two types of information for each detected object :

  - **Bounding Box Coordinates :** The first 4 values are for the location and size of the bounding box, which are :

    - **x :** The x-coordinate of the center of the bounding box

    - **y :** The y-coordinate of the center of the bounding box

    - **width :** The width of the bounding box

    - **height :** The height of the bounding box

  - **Class Probabilities :** The remaining 80 values correspond to the confidence scores for each class that the model is trained to identify. Each value represents the model's confidence that the detected object belongs to a corresponding class.

- **Number of Detections (3rd Dimension) :** The size of 8400 indicates that the model can detect a maximum of 8400 objects within a single image. This

large number is due to the multiple scales at which detections are performed, and multiple anchor boxes that the model uses to predict objects of various shapes and sizes.

The YOLOv8 model employs DIoU_NMS, an advanced version of non-maximum suppression (NMS), to process the vast array of potential detections (up to 8400 per image). DIoU_NMS improves upon traditional NMS by incorporating the concept of the "distance IoU." This metric not only evaluates the intersection over union (IoU) of bounding box overlaps but also the centrality of these boxes relative to the ground truth. By considering both the overlap and the distance between the centers of the predicted and ground truth bounding boxes, DIoU_NMS effectively enhances the selection process for the final object detections. This results in higher quality predictions and more precise suppression of redundant overlapping boxes, thereby increasing the overall accuracy of the object detection compared to the use of traditional NMS techniques. This refinement step is crucial in ensuring that the output of the model, as described earlier, represents the most accurate and relevant detections from the multitude of possibilities initially proposed by the network.

5 Enhancing Runway Detection Using Active Learning

5.1 Active Learning : An In-Depth Overview

5.1.1 Definition Active Learning

The remarkable success of deep learning models in recent years is largely attributable to the availability of extensive labeled training datasets. However, obtaining labeled data for new tasks, beyond existing benchmarks, is both challenging and expensive. Active learning can enhance the efficiency of labeling new data by identifying the most relevant images from a data pool, without the need for pre-existing labeling information.

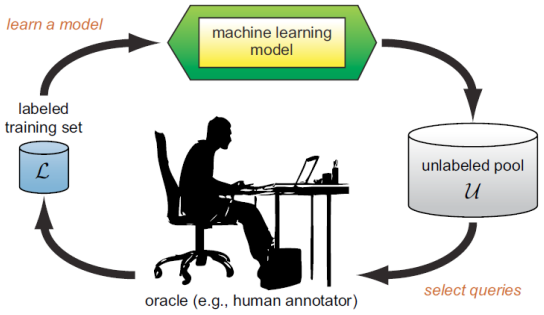

FIGURE 9 – Cycle of Active Learning [6]

Figure 9 illustrates the process of active learning in machine learning. A machine learning model is trained on a labeled dataset and then used to identify valuable queries from an unlabeled pool of data. These selected queries are labeled by a human annotator, referred to as an oracle, and the newly labeled instances are fed back into the training set.

5.1.2 Exploring Active Learning Scenarios

There are several different problem scenarios in which the learner may be able to ask queries. Figure 10 shows the three main settings that have been considered in the literature [6] :

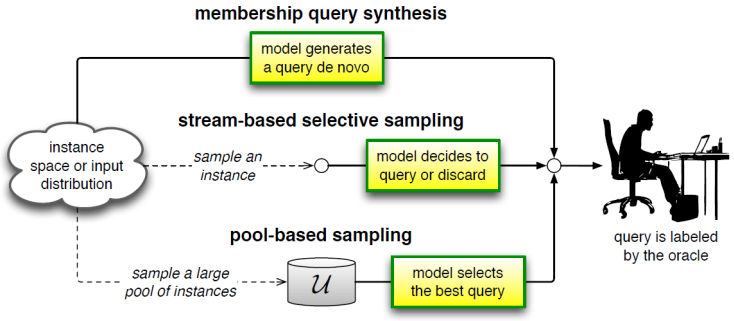

FIGURE 10 – Active Learning Scenarios [6]

Stream-Based Selective Sampling is an active learning technique tailored for data arriving in a continuous stream. In this method, data points are selected and labeled in real-time, allowing machine learning models to adapt dynamically to new information. This approach is crucial in applications where a constant influx of data requires immediate analysis and decision-making.

- Pros : It allows real time decision making.
- Cons : Limited performance due to separate decisions.

Pool-Based Sampling is a fundamental active learning strategy where a pool of unlabeled data is available, and the algorithm selects the most informative samples from this pool for labeling. Unlike stream-based methods, pool-based sampling allows algorithms to make batch selections from the available pool, making it suitable for scenarios where all data is accessible at once. Pool-based sampling methods often employ uncertainty measures or diversity criteria to choose the most valuable data points for labeling, maximizing the model's learning efficiency.

- Pros : Effective utilization of extensive unlabeled data resources.
- Cons : Incurs initial labeling expenses and inadequate for managing continuous data streams.

Membership Query Synthesis is an active learning approach where the learning algorithm generates queries by synthesizing instances that are particularly informative. Instead of relying solely on the available data, membership query synthesis allows the model to actively create hypothetical data points and request their labels. By strategically generating queries, this method aims to fill gaps in the model's knowledge, leading to more accurate predictions.

- Pros : Compatibility with problems where it is easy to generate new data instance.

- Cons : Elevated risk of data misidentification.

This study will utilize the Pool-Based Sampling approach, as there is already an existing data pool at our disposal.

5.2 YOLOv8's Features Extraction for Active Learning Strategies

In the Head of YOLOv8, bounding boxes, object classes, and objectness scores are predicted from the feature maps provided by the backbone. The head operates at various scales, specifically P3, P4, and P5, to detect objects of different sizes. These layers are pivotal as they offer rich information essential for identifying and localizing objects across a range of sizes. For each image, features from these three layers of the YOLO model – P3, P4, and P5 – are reshaped and transposed to align their dimensions. Focusing on these layers allows the model to capitalize on the unique strengths of each layer's scale in object detection. Subsequently, these features are vertically stacked to create a composite feature matrix for each image. The predictions made by YOLOv8, which include confidence scores for the detected objects, are utilized to weight the stacked features. This process effectively weights the features according to the confidence scores, generating a comprehensive descriptor for each image.
This will function as the input for the various active learning strategies outlined in the following section.

5.3 Data Selection in Active Learning

The strategies of active learning can be broadly categorized into two main types : uncertainty sampling and diversity sampling. Uncertainty sampling involves selecting samples for which the model has the lowest confidence in its predictions. The intuition behind this strategy is straightforward : by asking for labels for the most uncertain instances, the model rapidly acquires information it previously lacked, thus improving its learning efficiency.
On the other hand, diversity sampling is about choosing a set of samples that are diverse in feature space. This doesn't necessarily mean selecting the most uncertain samples, but rather samples that will best represent the underlying distribution of the data. The core idea is that by understanding the diverse characteristics of the dataset, the model can generalize better to new, unseen data.

5.3.1 Uncertainty Sampling Strategies

Active learning algorithms originally concentrated on the classification problem, aiming to improve the efficiency and accuracy of predictive models. In neural network-

based classification settings, where the final layer typically comprises a softmax activation, the outputs are often interpreted as probabilities. These probabilistic outputs are intended to provide a measure of confidence in the network's predictions, ideally reflecting the likelihood of each possible class. However, recent empirical findings indicate that softmax can produce overly confident predictions, particularly for unseen data instances. Highlighted in a series of studies [7] [8], this realization has intensified the research into methodologies for quantifying the uncertainty present in neural network outputs [7] [9] [10].
To address this issue, acquisition functions such as entropy, variation ratio, and margin are being utilized [9]. These metrics are instrumental in harnessing the predictive uncertainty of a network. By doing so, they serve as pivotal tools in the active learning process, allowing for a more informed and selective querying of labels. A noteworthy advancement was reported by Gat et al., who demonstrated a marked enhancement in image classification capabilities using an uncertainty-based acquisition function over a random baseline [9].
The forthcoming sections delve into the implementation of network uncertainty principles within the scope of active learning for classification tasks. We can easily extend each of these methods to an object detection setting by computing the variation ratio, margin or entropy for each descriptor that contains information about bounding boxes and confidence scores.

Variation-ratio (Least Confident) serves as an indicator of the softmax predictions' confidence deficit [9]. Our objective is to assign labels to instances based on the highest var ratio, giving lower priority to the bounding box's most confident class.

$$\text{var\_ratio}[x] := 1 - \max_y p(y|x, D_{\text{train}})$$

Margin sampling, also referred to as 1v2 sampling, aims to identify unlabeled data that can effectively refine the model's decision boundary. The equation bellow defines the margin for a bounding box. In essence, this metric endeavors to locate samples where the difference between the top two class probabilities in a bounding box is minimized.

$$v_{1vs2}(x) = \max_{c_1 \in K} \hat{p}(c_1|x) - \max_{c_2 \in K \setminus c_1} \hat{p}(c_2|x)$$

Brust et al. conducted experiments using the average, sum, and maximum methods to aggregate bounding box margins for each image. They discovered that these three aggregation techniques outperformed the baseline of random acquisition [11]. In our experiments, we will employ the maximum margin across all bounding boxes in an image as our chosen metric.

The predictive **entropy** of a classifier can be computed using the equation bellow [9], where the scaled log probability of the output belonging to each class is

summed. In the realm of deep learning classifiers, this calculation is usually performed at the softmax output layer of a deep network. The usual practice is to select samples with the highest entropy for querying.

$$H[y|x, D_{\text{train}}] := -\sum_{c} p(y = c|x, D_{\text{train}}) \log p(y = c|x, D_{\text{train}})$$

5.3.2 Diversity Sampling Strategy : The Coreset Approach

The coreset method, as defined by Sener et al. [12], is an approach to active learning tailored for batch sampling situations. The goal of the coreset selection problem is to identify a small subset from a large labeled dataset such that a model trained on this small subset performs comparably to one trained on the entire dataset. This method is designed to work without the use of labels and aims to minimize a rigorously defined bound between an average loss over any subset of the dataset and the remaining data points based on the geometry of the data points. This minimization is equivalent to solving the k-Center problem, for which Sener et al. adopt an efficient approximate solution to this combinatorial optimization problem.

The k-center problem is a well-known combinatorial optimization problem that arises in a variety of contexts, including clustering and facility location. Given a set of points and a positive integer k, the k-center problem seeks to determine k points (centers) among the given points such that the maximum distance of any point to its nearest center is minimized. In other words, the goal is to place k centers in such a way that the furthest any point has to travel to reach its nearest center is as small as possible.

Sener et al. evaluated their algorithm on CIFAR [13] and SVHN [14] datasets, examining its performance for both fully-supervised and weakly-supervised models. The results demonstrated that the coreset method surpassed other baselines, with particularly strong performance in weakly-supervised settings due to improved feature learning. However, its effectiveness was slightly reduced for the CIFAR-100 [13] dataset, a variation attributed to the increase in the number of classes.

In this study, we apply the k-center problem, using the descriptors of images as input.

6 Practical Considerations and Strategies

6.1 Integrating Active Learning with Deep Learning Models

Using Active Learning with Deep Learning presents several challenges, including :

Complex Deep Learning Model Architecture : The intricate architecture of deep learning models adds complexity to the integration with Active Learning. In a classification model, uncertainty measurement based on class probability distribution facilitates query selection for labeling. However, in the case of yolov8, which outputs bounding box coordinates and the probability of runway presence, employing uncertainty or diversity strategies becomes challenging.

Acquisition Function Limitations : Existing acquisition functions typically propose a single best image for labeling, rather than a batch of images. While suggesting the top K images based on an information measure is an option, issues such as image similarity or redundancy may arise, potentially hindering improvements in the model's training performance.

Catastrophic Forgetting (Knowledge Erosion) : As the model learns from new incoming data, there is a risk of weakening the representation of older data. The model should ideally retain and build upon its knowledge over time to prevent knowledge erosion.

Data Quantity Dilemma : Deep learning models thrive on extensive training data, while active learning aims to minimize the amount needed. Striking a balance between introducing/exploring new examples and exploiting existing knowledge can be challenging.

Model Initialization Impact : The performance of a model, particularly when it begins with a random initialization, may necessitate additional labeled data to achieve acceptable accuracy. This underscores the importance of addressing model initialization to enhance the efficiency of integrating Active Learning with Deep Learning. Additionally, factors such as repeatability and consistency, which can be influenced by data augmentation and similar techniques, also play a critical role in the process.

6.2 Impact of Weight Re-Initialization

In addition to differences in query approaches, researchers have employed various methods in the training process of their networks. Notably, some researchers opted to re-initialize the weights of their networks when incorporating new data [10] [15] [9], while others chose not to do so [16] [11]. Our literature review reveals a gap in studies examining the essentiality of network weight re-initialization in the context of active learning for object detection. Consequently, this study aims to investigate the effects of weight re-initialization on deep object detectors when new data is acquired.

7 Experimental Results

Our evaluation process involves testing various strategies and benchmarking their performance against the baseline method, which is random sampling. This is executed in a pool-based scenario across 20 rounds or iterations. Initially, we begin with a modestly-sized labeled dataset comprising 500 images, randomly chosen from the input dataset. In every iteration, we select an additional 500 images employing one of several active learning strategies (such as coreset, entropy, margin, or var_ratio). Each experiment consistently uses the same strategy throughout. Subsequently, a YOLOv8 model is trained from scratch for each iteration, utilizing the initial dataset augmented with the newly selected images.
To thoroughly assess the effectiveness of these strategies, we calculate different metrics (Annex A in page 38) in two specific ways. We average these metrics across the 20 epochs of training. This averaging approach gives us a comprehensive view of the model's performance over the entire training period, offering insights into the stability and consistency of the learning process under each active learning strategy. We also give special consideration to the metrics obtained in the last epoch of training. This focus on the final epoch's metrics is crucial, as it provides an understanding of the model's performance at the point of convergence. It helps in evaluating how well the model has adapted to the cumulative learning over all iterations, reflecting its potential for generalization and robustness.

7.1 Performance of Active Learning Strategies

Firstly, **data augmentation is consistently enabled** in our experiments to evaluate its impact. We assess the effectiveness of various active learning strategies, particularly focusing on coreset and entropy-based selection. The latter serves as a representative of uncertainty-based methods, and these strategies are compared against the baseline of random sampling.

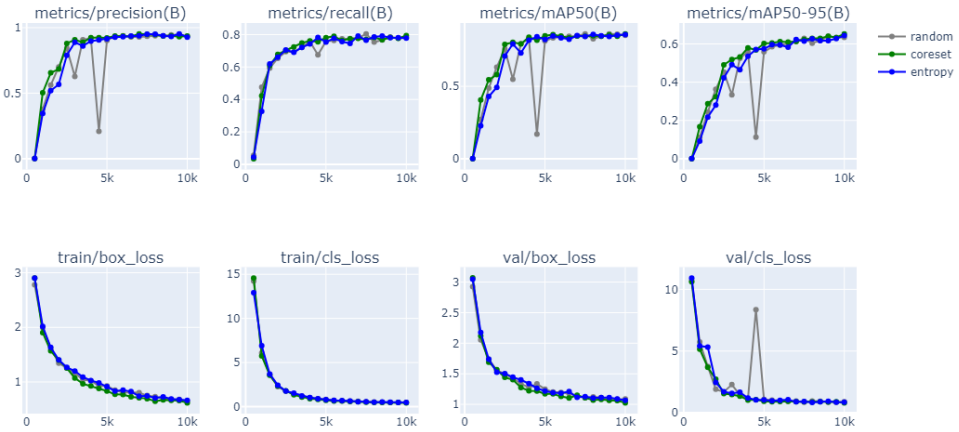

FIGURE 11 – Random vs Coreset vs Entropy : using last epoch

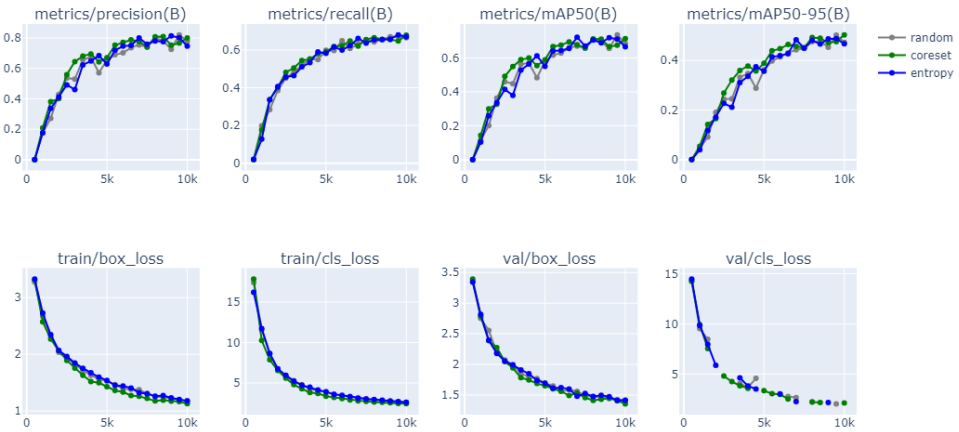

FIGURE 12 – Random vs Coreset vs Entropy : using mean on epochs

Both **coreset** and **entropy** strategies demonstrate better performance over the random strategy in terms of the precision, recall, and mAP metrics. This suggests that both strategies are more effective at selecting informative samples for training the model compared to random sampling. The entropy strategy, in particular, shows a consistent lead over random across all performance metrics, underscoring the value of uncertainty-based methods in active learning for object detection tasks. However, the differences in the loss metrics are less pronounced, indicating that all strategies eventually lead to a similar level of model optimization.
The **entropy** strategy seems to be the most effective, both in terms of average (fi-

gure 12) and final-epoch (figure 11) performance. The coreset strategy, while better than random, shows a less pronounced improvement.
However, the differences between the strategies are marginal, and with data augmentation not being turned off, it could indeed influence the results. Data augmentation can introduce variability and complexity into the training data, which might enhance the model's ability to generalize but also can make it harder to directly attribute improvements in performance to the active learning strategies themselves.
Secondly, with **data augmentation** deactivated, our attention is centered on the metrics averaged across the 20 epochs of training (refer to Annex B in page 40) for the results pertaining to the metrics from the final epoch of training) :

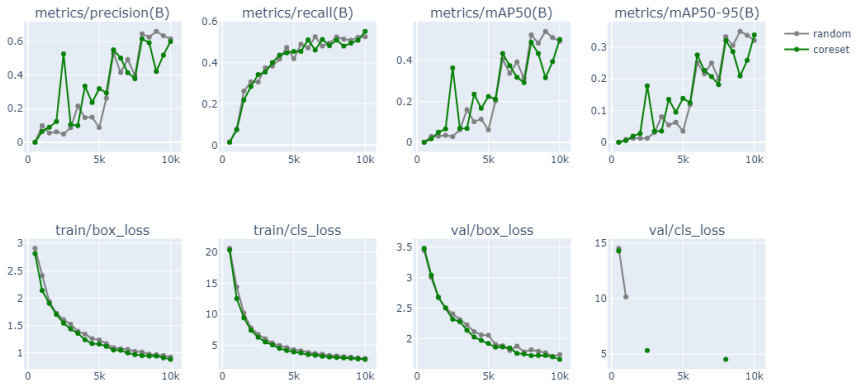

FIGURE 13 – Random vs Coreset

FIGURE 14 – Random vs Entropy

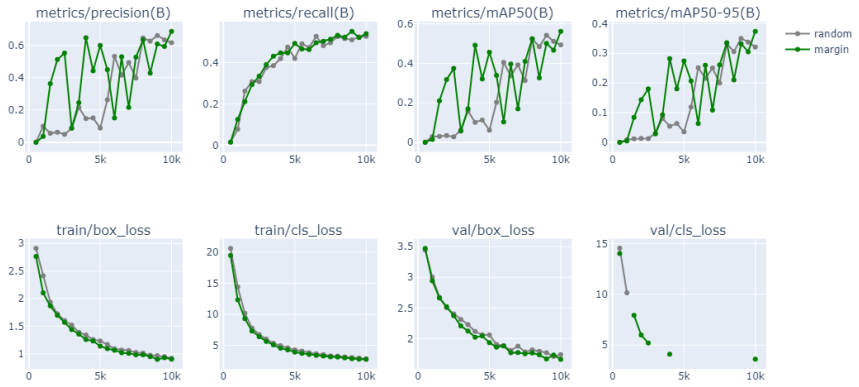

FIGURE 15 – Random vs Margin

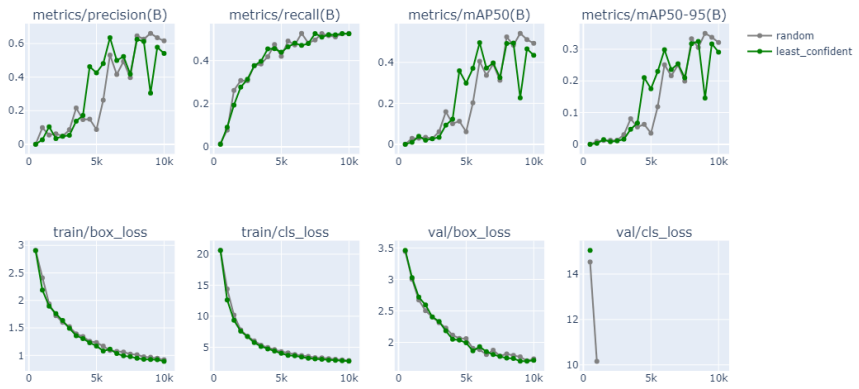

FIGURE 16 – Random vs Least Confident

As shown in figure 13, the **Coreset** strategy demonstrates a generally better sample efficiency with higher peaks in precision and mean Average Precision (mAP) scores, indicating a superior ability to identify relevant samples for the model to learn from. While the recall rates of Coreset and random strategies are closely matched, the slight edge that Coreset maintains suggests it might be better at ensuring the model detects relevant objects. The loss metrics between Coreset and random converge similarly, which points to a similar model optimization level over time, despite the variations in sample selection.

Entropy shows promise, in figure 14, with its often higher precision and mAP scores, suggesting that it is effective at selecting informative samples that lead to more ac-

curate detections. The recall performance is comparable or slightly better than the random strategy, reinforcing the effectiveness of entropy-based selection. However, the similar convergence patterns in loss metrics between entropy and random strategies indicate that the entropy strategy's benefits might be more pronounced in the early stages of training.

As for the **Least Confident** strategy in figure 16, while showing significant volatility, does not consistently outperform the random strategy in precision. Its recall performance is comparable, and there are periods where it achieves better mAP scores. However, the spikes observed in validation loss could suggest moments of overfitting, indicating that this strategy might benefit from further tuning to stabilize its performance.

Margin's performance (figure 15) closely resembles the random strategy with moments of outperformance in precision and mAP metrics. This indicates that while margin-based selection can sometimes be beneficial, it doesn't consistently provide a clear advantage over random sampling. As with the other strategies, the loss metrics eventually show a similar downward trend, suggesting that the model's learning process ultimately smooths out the differences in active learning strategy efficiencies.

In summary, while these active learning strategies exhibit some level of improvement over random sampling in various metrics, the entropy strategy stands out with more consistent performance enhancements, particularly in precision and mAP scores. The convergence of loss metrics across all strategies suggests that differences in active learning impact may diminish as the model's exposure to data increases over time.

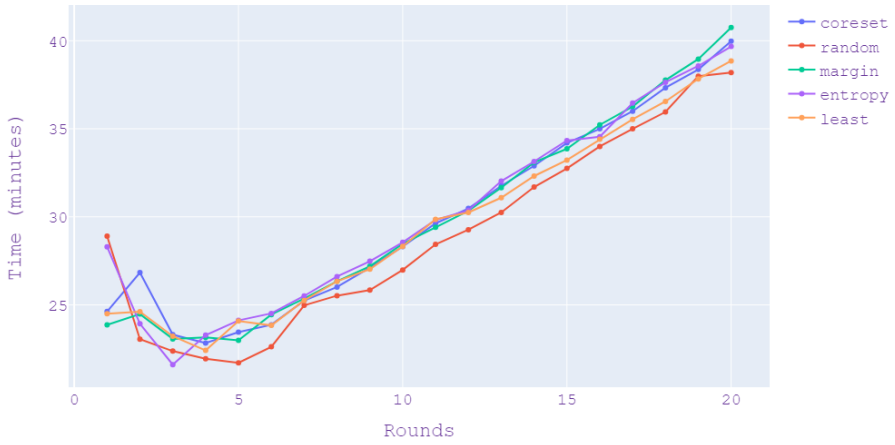

FIGURE 17 – Comparative Time Analysis Across Strategies

As the dataset grows with each round, a corresponding rise in training time is anticipated (figure 17) ; however, the trend of various active learning strategies converging to similar training durations in the latter rounds implies that the method of sample selection has a minimal impact on the length of training time. Early fluctuations in training times can be attributed to the computational demands inherent in each strategy's approach to selecting new data — strategies with more intricate selection processes might have a slower start. Ultimately, the convergence of training times suggests that the time required for incremental training stabilizes and becomes increasingly uniform across strategies as the model progresses through successive training rounds.

7.2 Weights Reinitialization

When comparing the Coreset and Entropy strategies using average metrics and metrics from the last epoch, distinct patterns emerge, revealing the effectiveness of incremental learning versus training new models from scratch.

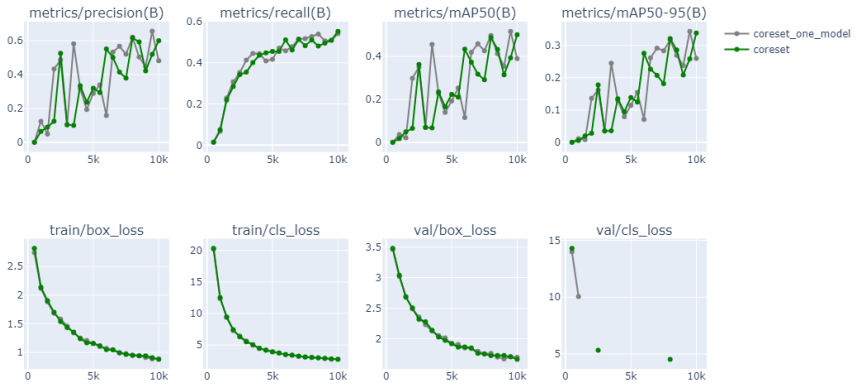

FIGURE 18 – Coreset vs Coreset using one model

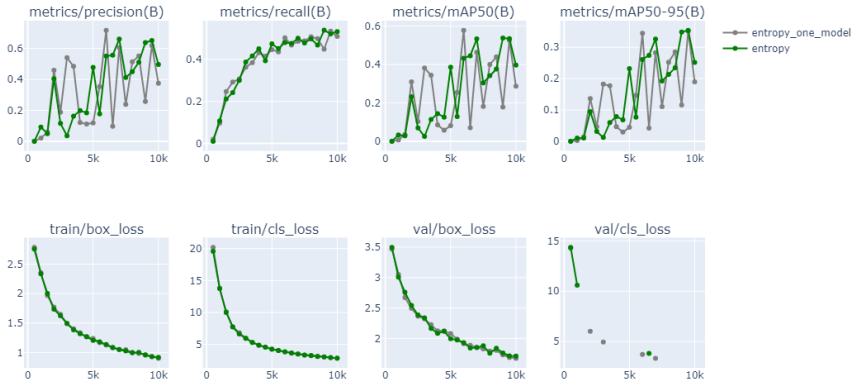

FIGURE 19 – Entropy vs Entropy using one model

With the average metrics (figures 18, 19) the **Coreset** strategy shows a notable benefit when using the 'coreset_one_model' approach. It consistently attains higher precision, recall, and mAP scores across training iterations. This improvement suggests that a model that incrementally learns from new data can better refine its understanding of object detection, leading to more accurate predictions. The loss metrics between the 'coreset_one_model' and 'coreset' are comparable, with both converging similarly, indicating that the overall capacity for learning does not differ significantly between the two methods. Yet, the more gradual and stable progression of the 'coreset_one_model' approach underscores the advantages of cumulative learning.

The **Entropy** strategy also displays advantages with the 'entropy_one_model' approach, which typically outperforms training a new model for each iteration. The enhanced precision, recall, and mAP scores, although with a less pronounced difference than the Coreset strategy, imply that the Entropy strategy can benefit from building on previous learning but is not as heavily reliant on it as Coreset might be. The similar loss trends further suggest that the fundamental learning process remains consistent regardless of the approach.

FIGURE 20 – Random vs Coreset

FIGURE 21 – Random vs Entropy

Looking at the metrics from the last epoch (figures 20, 21), we observe that the '**coreset**_one_model' approach presents a somewhat erratic behavior in precision and mAP, contrasting with the more consistent yet slightly lower performance of

training new models. Both approaches demonstrate comparable recall rates, suggesting a similar capability to identify the relevant objects within the data. However, the 'coreset_one_model' tends to show a steadier decline in loss metrics, pointing towards a more uniform learning experience.

For the **Entropy** strategy, the 'entropy_one_model' method generally achieves better precision and mAP values in the last epoch, signifying an effective use of accumulated knowledge. Yet, this approach experiences greater volatility, hinting at possible overfitting to specific data batches. This is further evidenced by the occasional spikes in validation loss, which might raise concerns about the model's stability when confronted with new data.

In summary, the 'one_model' approaches for both strategies demonstrate the potential benefits of continuous learning, with higher performance peaks suggesting more effective optimization from past learning experiences. However, the increased volatility, particularly with the Entropy strategy, may indicate a trade-off with model stability. On the other hand, training new models from scratch each iteration yields a more consistent performance, albeit with slightly lower peaks. This consistency could translate into better generalization when the models are applied to unseen data. Thus, the choice between these approaches may depend on the desired balance between immediate performance and long-term robustness.

8 Conclusion and Discussion

In their comprehensive study, David Lowell et al. [17] examined various standard active learning methods across different models, domains, and acquisition functions for two primary NLP tasks : text classification and sequence tagging. The study also evaluated the effectiveness of using training datasets, actively sampled by an acquisition model, with a different successor model. This aspect is crucial considering the rapid evolution of ML models and the longer lifespan of annotated datasets, which often outlive the models they were initially intended for.
The findings from Lowell and his team [17] highlight the unpredictable nature of active learning's performance. While certain combinations of acquisition functions, models, tasks, and domains might demonstrate effectiveness, predicting this success in advance remains a challenge. The study underscores that without gathering a large amount of random sampled data, it's impossible to retrospectively assess the relative success of active learning. This approach, however, contradicts active learning's primary purpose. Moreover, even if a random sample is used early in the active learning cycle as a diagnostic tool, its early success doesn't guarantee similar results later in the cycle, as illustrated in Figure 22.

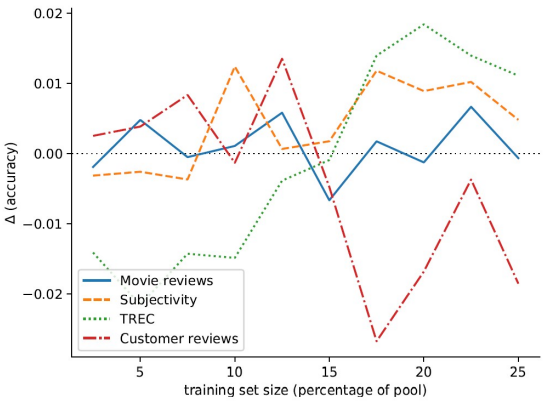

FIGURE 22 – Performance of AL relative to random across datasets

A significant concern identified in the study [17] is that even in cases where active learning is successful, the training set obtained is closely tied to the model used for its acquisition. When these datasets are employed to train successor models, the performance often falls short of what could be achieved with an equivalently sized random sample. The study does note that results for Named Entity Recognition (NER) were more favorable compared to text classification, aligning with previous research in this area.
In summary, the research by David Lowell et al. [17] indicates that the performance of individual active acquisition functions can vary greatly across different datasets and domains. While active learning can sometimes outperform random sampling,

these advantages are typically marginal and inconsistent. Furthermore, the datasets acquired through active learning may not be as beneficial for training subsequent models, raising significant concerns about the practical effectiveness of active learning in the field of machine learning.

Conclusion

In concluding this report, we can synthesize the extensive experimental work and analyses performed in enhancing runway detection using active learning strategies. The integration of active learning with the deep object detection model YOLOv8 has been a complex yet insightful endeavor. The key findings indicate that while strategies like coreset and entropy can outperform random sampling in selecting informative samples, their advantages tend to be marginal and nuanced.

The study has demonstrated that the continuous refinement of a single model via incremental learning can effectively leverage prior knowledge and lead to higher performance metrics. However, this approach is not without its trade-offs, as increased volatility in performance metrics suggests potential overfitting issues. In contrast, training a new model from scratch at each iteration offers more consistency, potentially translating into better generalization on unseen data.

Moreover, the convergence of training times across different active learning strategies suggests that the incremental time required for retraining becomes more predictable and less dependent on the selection method, especially in later training rounds. This finding is particularly relevant in practical applications where training efficiency is paramount.

The experiments have also highlighted the dual nature of deep learning models that thrive on extensive training data yet must be refined efficiently through active learning to minimize data requirements. Balancing the exploration of new examples and the exploitation of existing knowledge emerges as a central theme in the efficient application of active learning strategies.

As we look to the future, the insights gained from this research can guide the development of more robust and effective active learning frameworks. The aim will be to create models that not only achieve high performance quickly but can also maintain stability and generalization capabilities in the face of evolving data landscapes. The nuanced understanding of when and how to apply different active learning strategies will be crucial in advancing the field of autonomous object detection and beyond.

9 References

[1] Glenn JOCHER, Ayush CHAURASIA et Jing QIU. *Ultralytics YOLO*. Version 8.0.0. Available from Ultralytics. 10 jan. 2023. URL : https://github.com/ultralytics/ultralytics.

[2] *MS COCO : Common Objects in Context*. Accessed : 2024-01-16. URL : http://cocodataset.org.

[3] Olga RUSSAKOVSKY et al. "ImageNet Large Scale Visual Recognition Challenge". In : *International Journal of Computer Vision* 115.3 (2015), p. 211-252.

[4] Juan TERVEN et Diana-Margarita CORDOVA-ESPARZA. *A Comprehensive Review of YOLO : From YOLOv1 to YOLOv8 and Beyond*. Avr. 2023.

[5] ULTRALYTICS. *Ultralytics YOLO*. https://github.com/ultralytics/ultralytics. Retrieved January 19, 2024. 2023.

[6] Burr SETTLES. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison, 2009. URL : https://minds.wisconsin.edu/bitstream/handle/1793/60660/TR1648.pdf?sequence=1.

[7] Balaji LAKSHMINARAYANAN, Alexander PRITZEL et Charles BLUNDELL. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In : *Advances in Neural Information Processing Systems 30*. Sous la dir. d'Isabelle GUYON et al. Accessed : 2019-10-06. Curran Associates, Inc., 2017, p. 6402-6413. URL : http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.pdf.

[8] Yarin GAL. *Uncertainty in Deep Learning*. University of Cambridge. PhD Thesis, p. 174, Available at : [URL], Accessed : 2019-10-06. 2016.

[9] Yarin GAL, Riashat ISLAM et Zoubin GHAHRAMANI. "Deep Bayesian Active Learning with Image Data". In : *arXiv :1703.02910 [cs, stat]* (mars 2017). Accessed : 2019-10-05. URL : http://arxiv.org/abs/1703.02910.

[10] Di FENG et al. "Deep Active Learning for Efficient Training of a LiDAR 3D Object Detector". In : *arXiv :1901.10609 [cs]* (jan. 2019). Accessed : 2019-10-05. URL : http://arxiv.org/abs/1901.10609.

[11] Constantin-Alexander BRUST, Christoph KÄDING et Joachim DENZLER. "Active Learning for Deep Object Detection". In : *arXiv :1809.09875 [cs]* (sept. 2018). Accessed : 2019-10-05. URL : http://arxiv.org/abs/1809.09875.

[12] Ozan SENER et Silvio SAVARESE. "Active Learning for Convolutional Neural Networks : A Core-Set Approach". In : *International Conference on Learning Representations*. 2018. URL : https://openreview.net/forum?id=H1aIuk-RW.

[13] Alex KRIZHEVSKY, Vinod NAIR et Geoffrey HINTON. *The CIFAR-10 Dataset*. https://www.cs.toronto.edu/~kriz/cifar.html. Accessed : 2019-10-05. 2014.

[14] *Street View House Numbers (SVHN) Dataset*. http://ufldl.stanford.edu/housenumbers/. Accessed : [Access Date].

[15] Chung-Ching KAO et al. "Localization-Aware Active Learning for Object Detection". In : *arXiv :1801.05124 [cs]* (jan. 2018). Accessed : 2019-10-05. URL : http://arxiv.org/abs/1801.05124.

[16] S. ROY. *Deep active learning for object detection*. p. 12, Available at : [URL], Accessed : 2019-10-05. 2018.

[17] David LOWELL, Zachary C. LIPTON et Byron C. WALLACE. *Practical Obstacles to Deploying Active Learning*. 2019. arXiv : 1807.04801 [cs.LG].

Annex A : Metrics Definitions

Here's a definition of each metric used to evaluate the performance of YOLOv8 :

Precision (metrics/precision(B)) : Precision measures the accuracy of the positive predictions. It is the proportion of true positive detections (correctly detected objects) over the total number of positive predictions (true positives + false positives). High precision indicates a lower rate of false positives.

Recall (metrics/recall(B)) : Recall, or Sensitivity, measures the model's ability to correctly identify all relevant objects. It is the proportion of true positive detections over the total number of actual positives (true positives + false negatives). High recall indicates that the model is good at detecting most of the objects of interest.

mAP50 (metrics/mAP50(B)) : Mean Average Precision at 50% Intersection over Union (IoU). mAP50 is a popular metric for object detection models. It calculates the average precision at 50% IoU threshold. IoU is a measure of overlap between the predicted bounding box and the ground truth. At 50% IoU, a prediction is considered correct if the overlap is 50% or more.

mAP50-95 (metrics/mAP50-95(B)) : This is an extension of mAP50. It calculates the mean Average Precision over multiple IoU thresholds, from 50% to 95% in increments of 5%. It gives a more comprehensive view of the model's performance across various levels of strictness in object localization.

Train Box Loss (train/box_loss) : During training, this metric measures the loss related to the bounding box predictions. It quantifies how much the predicted bounding boxes deviate from the ground truth boxes. Lower box loss indicates better performance in predicting the location and size of the objects.

Train Class Loss (train/cls_loss) : This represents the loss related to the classification of objects within the training set. It measures how accurately the model classifies the objects in the bounding boxes. Lower class loss indicates higher accuracy in predicting the correct class labels for detected objects.

Validation Box Loss (val/box_loss) : Similar to training box loss, but calculated on the validation set. It helps in evaluating how well the model generalizes its ability to predict bounding boxes on unseen data.

Validation Class Loss (val/cls_loss) : This is the class loss calculated on the validation set. It indicates the model's performance in classifying objects on data

not seen during training, hence providing insight into the model's generalization capability.

These metrics collectively provide a comprehensive assessment of an object detection model's performance, covering aspects of localization (box loss), classification (class loss), and overall detection accuracy (precision, recall, mAP).

Annex B : Plots

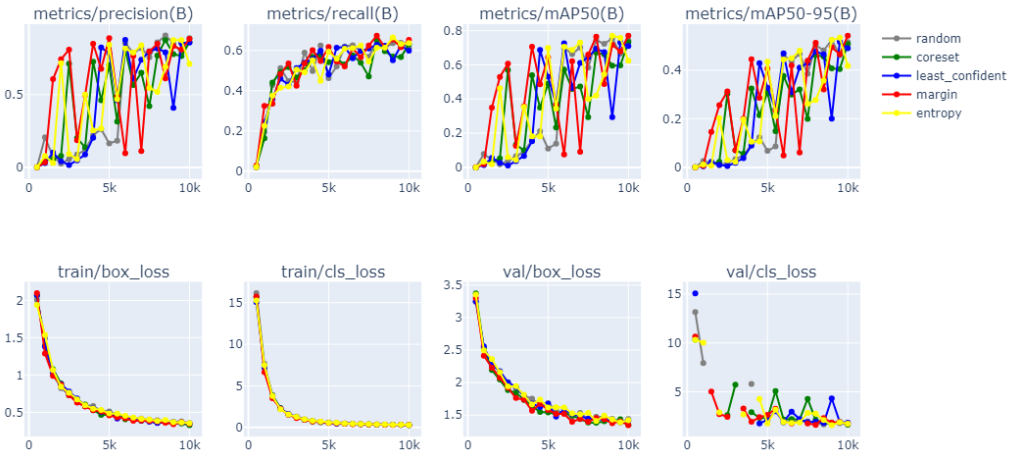

FIGURE 23 – Random vs All : metrics on last epoch

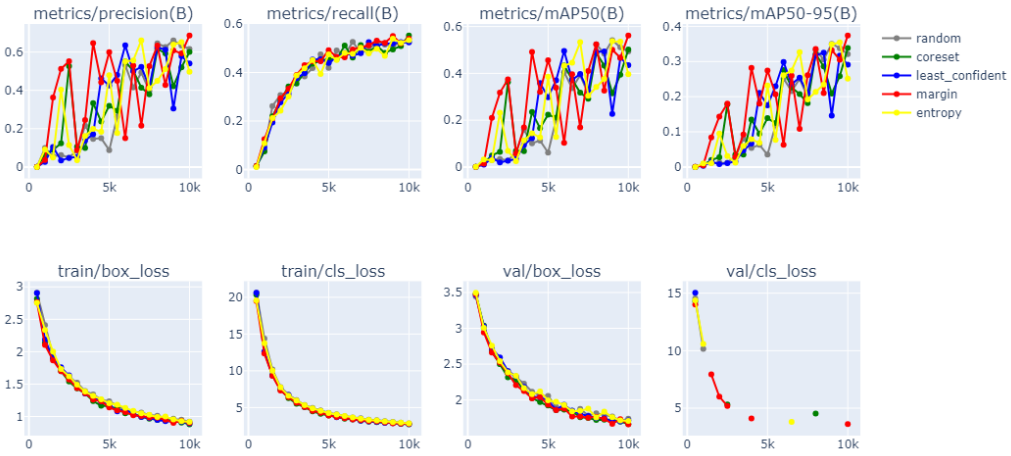

FIGURE 24 – Random vs All : averaged metrics

Different results with metrics computed on the last epoch of training and data augmentation disabled.

FIGURE 25 – Random vs Coreset

FIGURE 26 – Random vs Entropy

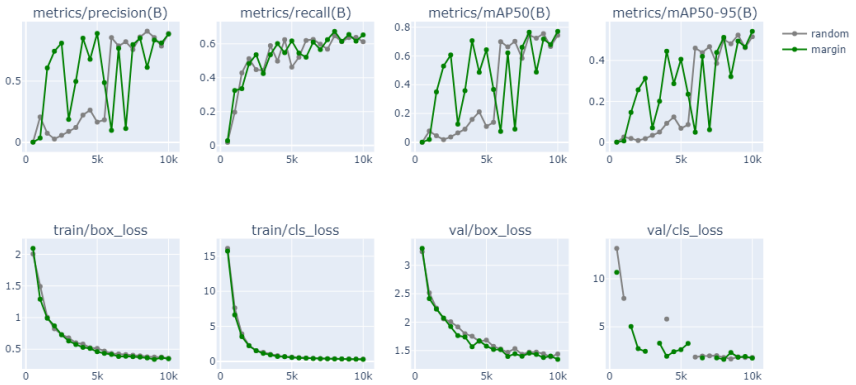

FIGURE 27 – Random vs Margin

FIGURE 28 – Random vs Least Confident