Mixing Real and Synthetic Data in Neural Network Training: A Case Study Regarding Runway Detection

Gustavo H. Daniel*† and Juliano E. C. Cruz‡
Embraer, São José dos Campos, São Paulo, 12227-901, Brazil

Marcos R. O. A. Maximo§
Aeronautics Institute of Technology (ITA), São José dos Campos, São Paulo, 12231-200, Brazil

Runway detection is an essential part of visual-only automatic landing systems. Training deep neural networks for this task requires large and diverse datasets that are often costly and difficult to obtain. The present study systematically evaluates the effect of mixing real and synthetic images on the performance of a YOLO-based neural network for corner point extraction. Dataset compositions were varied by progressively replacing real images with synthetic ones to determine the optimal balance between data sources and model accuracy. Incorporating 10–30% real images within a mixed dataset was found to yield performance comparable to using exclusively real data, while the inclusion of synthetic images significantly improved generalization to diverse scenarios. These results provide practical guidelines for constructing datasets in runway detection and inform future developments in neural network training for autonomous aviation systems.

I. Introduction

Deep neural networks have demonstrated remarkable capabilities across a wide range of applications, from image recognition to autonomous systems [1–3]. However, one of their most significant limitations is the need for large, diverse, and high-quality datasets to achieve robust performance [4]. Acquiring such datasets from real-world environments can be both time-consuming and costly, particularly in domains where data collection involves complex logistics or safety concerns [4]. As a result, many researchers turn to simulated environments to generate synthetic data, which offers a scalable and cost-effective alternative [5–7]. This approach is especially relevant in the context of runway detection or segmentation for autonomous aircraft landing. Collecting real-world data for this task typically requires repeated flights in different regions, using different aerodromes, and under various conditions, which is not only expensive but also operationally restrictive [8, 9]. While synthetic images generated through simulation can help mitigate these challenges, they often fail to capture the full variability, complexity, and unpredictability of real-world scenarios. Therefore, combining real and synthetic data has emerged as a promising strategy to balance cost-efficiency with model generalization [4].

The importance of improving landing systems is underscored by aviation safety statistics. The final approach and landing phases are the most critical stages of flight. Together, they account for approximately 56% of all fatal aviation accidents [10]. Moreover, human error—especially pilot error—remains a leading cause, contributing to nearly 80% of these incidents [11].

Autonomous landing systems offer a compelling solution by reducing pilot workload and minimizing the risk of human error. Such systems can be implemented using a variety of technologies, including the Instrument Landing System (ILS) [12], Global Navigation Satellite Systems (GNSS), and computer vision-based approaches. Each of these methods has its own advantages and limitations. For instance, GNSS-based auto-landing requires high-precision augmentation systems such as the Ground-Based Augmentation System (GBAS) to achieve the necessary accuracy, and its deployment involves significant infrastructure investment.[13]. Similarly, ILS-based systems depend on ground-based transmitters, which are costly to install and maintain. As a result, many aerodromes —particularly smaller or remote ones— do not have the necessary equipment to support these systems [9, 13]. In contrast, computer vision-based auto-landing

*Engineer, Research & Technology Department, Embraer, Av. Brg. Faria Lima, 2170.
†Student, Aeronautics Institute of Technology (ITA), Praça Marechal Eduardo Gomes, 50.
‡Engineer, Research & Technology Department, Embraer, Av. Brg. Faria Lima, 2170.
§Professor, Autonomous Computational Systems Laboratory (LAB-SCA), Computer Science Division, Aeronautics Institute of Technology (ITA), Praça Marechal Eduardo Gomes, 50.

presents a flexible and scalable alternative. By relying on onboard cameras and image processing algorithms, this approach can be implemented on virtually any aircraft, regardless of the airport's infrastructure. It holds the potential to democratize access to autonomous landing capabilities and improve safety across a broader range of operational environments. However, this method also faces challenges—most notably, the requirement for a clear visual of the runway during landing. Adverse weather conditions, such as fog or heavy rain, can obscure visibility and compromise system performance.

This paper addresses the challenge of data scarcity in runway detection by quantifying the performance trade-off between real and synthetic data, training a YOLOv11 model for the task. Our main contribution is the empirical determination of the optimal data composition in our dataset, which demonstrated that combining a minimal proportion of real-world images (up to 30%) is sufficient to achieve high performance, thereby offering insights to help future development of autonomous landing technologies.

The remainder of this paper is organized as follows. Section II presents a review of studies that address the need for a large quality dataset to train neural networks or that have the main objective related to runway detection and segmentation. Section III describes the methodology used in the present paper regarding the ratio of real and synthetic images in the dataset. Section IV presents and analyzes the results. Section V presents the conclusions and future directions for research.

II. Literature Review

For the specific task of runway detection and segmentation, gathering images is expensive considering the costs of flying an airplane, especially given the variety of runways, meteorological conditions, and time of the day needed for a robust dataset. To solve or at least mitigate such a problem, researchers use satellite imagery, videos from the web, and simulators (e.g. FlightGear and X-Plane), as will be shown in the following paragraphs.

Akbar et al. [14] used satellite images taken from Google Earth. In their study, the authors trained a classifier that uses ResNet50 as backbone to find out if there is a runway in the image and a runway detection algorithm that uses Hough Transform [15] as well as Line Segment Detector [16], based on the study by Grompone von Gioi et al. [16], to detect edges and lines to produce the bounding box of the runway. With the bounding box, the Mask R-CNN [17] is used to make the segmentation of the runway. Using this method, the authors achieved an mean average precision (mAP) of 94% with a threshold of 0.5-0.6. Although the research presented a reasonable performance, the dataset used to train consists of 700 images, which is typically considered small for deep learning methods.

Khelifi et al. [18] used the Airport Data and Information Portal (ADIP) provided by the Federal Aviation Administration (FAA) to get the location of the airports and the Google Static Maps API to take the images. After taking the images, the authors manually annotated the airport runways in their dataset. The study employed Faster R-CNN [19] with Resnet-101 and Resnet-X152. To improve the performance, they incorporated simple data augmentation, namely random brightness, contrast, saturation, lightning, rotation, flip, and gray scale transformation. They also performed transfer learning importing pre-trained weights on COCO [20], hyperparameter fine-tuning, as well as negative samples of highways due to their resemblance to runways. The best result obtained was 99% precision, 88% recall, and 76% mAP, with a 0.5 IoU threshold, using the ResNet-X152 backbone.

However, the main problems with satellite imagery are the angle of the images and the inability to simulate different weather conditions [21, 22]. Unlike satellite imagery, onboard cameras capture the runway from the aircraft's perspective during landing, resulting in images that are influenced by weather conditions and varying environmental factors such as sky appearance and illumination. To avoid these problems, some authors have used images from flight simulators.

Tsapparellas et al. [2] developed a real-time vision-based system for runway detection, enabling Unmanned Aerial Vehicle (UAV) landing. It uses an algorithm based on Scale Invariant Feature Transform (SIFT) [23] and Speeded-Up Robust Feature Transform (SURF) [24] for extracting and matching features from the runway images. Then the authors used the Channel and Spatial Reliability Tracker [25] to detect the bounding box of the runway in the following images, and after a certain number of images, the detection and matching algorithm is used again to refine the runway bounding box for improved accuracy. The performance validation for such algorithms was performed using a combination of simulated images from X-Plane 11 and real images from a UAV, although the specific proportion used remains undisclosed. The study results show that the average accuracy is 94.89% in clear weather conditions.

Chen et al. [26] also used X-Plane for runway detection and segmentation and developed the Benchmark for Airport Runway Segmentation (BARS), which is one of the largest publicly available datasets in number of annotated images for runway segmentation, with a total of 10,256 images. The dataset includes a variety of scenes and has a large variation obtained in different weather conditions and times of day. However, scenarios such as foggy and hazy days are not

included due to the simulator's limitations. The drawback of this study is that the generated dataset uses only synthetic images, which may limit its performance in real-world scenarios.

Lima et al. [6] created a synthetic dataset using the FlightGear simulator to train a NN model for runway keypoint detection and used real images for validation. To generate the dataset, the authors generated the images with green spheres at each corner of the runway inside the simulation and compared screenshots with and without these green spheres to acquire their location. However, the author did not disclose information about the weather and time of the day used in the simulation. In addition, the study also employed an algorithm (Perspective N-point) in order to infer the aircraft pose with the runway keypoints. The study achieved a high Average Precision (AP) of 96%, at a 0.5 threshold for the IoU, on the simulated image dataset. However, performance on real-world images was comparatively lower, likely due to a domain gap between real and synthetic images, which hinders definitive conclusions. Based on these findings, the author recommends leveraging synthetic datasets for initial training phases. This approach can reduce reliance on real images and significantly decrease the time and cost associated with manual annotation, thereby enabling more efficient and scalable NN training.

Apart from dedicated flight simulators, other software might generate synthetic images. Ducoffe et al. [9] used Google Earth Studio to create a synthetic dataset. Then the authors created a dataset called Landing Approach Runway Detection (LARD), which also used images from real commercial aircraft landing videos available on the internet, which is different from Chen et al.[26] that only used synthetic images. The dataset comprises a training set of 14,443 synthetic images and a test set of 2,221 synthetic images, as well as 103 hand-labeled pictures from real landing footage. Consequently, only a small portion of the dataset is derived from real landing videos. Another limitation of this dataset is the conditions of the images, where there is only one runway per image, the runway is fully visible, and there are no adverse weather conditions.

Quessy et al. [27] trained a CNN using a dataset created using Unreal Engine 4 with the Microsoft AirSim plugin to make semantic segmentation of runway images. However, the dataset was composed of only runways from Lydd airport, which may negatively affect the model's performance on generalization, since other aerodromes will have different characteristics, especially the surrounding environment. In the second part of the study, the segmentation model was tested on real images from the same airport. A significant performance degradation was observed. To address this issue, the authors suggest the use of fine-tuning using real-world images.

Wang et al. [28] proposed an algorithm for runway detection for landing UAVs at night. The dataset is composed of a fusion of visible and infrared images acquired during nighttime in the FlightGear simulator. The model employed in the study used an improved version of Faster R-CNN and outputs the relative position of the UAV and the runway. The authors simulated three different airports with a total of 1,500 visible and 1,500 infrared images in the dataset, achieving an AP of 84.37%.

Li et al. [29] developed the YOLO-RWY, a new deep learning model built upon the YOLOv8 framework, it incorporates several enhancements optimizing performance for runway detection. The authors conducted experiments using LARD [9], and the new architecture achieved up to 40.2% performance increase in accuracy. The scores of mAP50-95 are 0.760 for the synthetic test set and 0.578 for the real test set.

The need for a large quality dataset to train NNs effectively is not exclusive to runway detection and has already been addressed by several authors [30, 31]. Despite this need, the available datasets for many computer vision applications, including runway detection, are scarce, commonly due to the costs of gathering the data. To solve this problem, different authors use different approaches, such as pre-trained models [30], Generative Adversarial Networks (GANs) [1, 32, 33], or even manually created with graphic software or game engines [5, 34].

A common practice to mitigate the need for a large quality dataset is to pretrain the NN on another larger and more general dataset. Fatima Ezzahra et al. [30] evaluated the use of transfer learning to train a CNN for traffic sign detection. This technique can leverage knowledge from a large dataset to improve performance on a smaller one. The paper compared three popular transfer learning-based CNN architectures, namely ResNet, VGGNet, and MobileNet. All architectures were trained using the ImageNet dataset, which is a large, general-purpose dataset, and then all pre-trained models were fine-tuned using a dataset with traffic signs from Germany. The model with the best accuracy was MobileNetV2 with 77.60%. Although the study did not disclose the size of the traffic sign dataset and the fact that there is still room for improvement in accuracy, the study presented a realistic option to solve the problem of scarce datasets.

Replacement of real images with synthetic ones is a solution that has been investigated by Burdorf et al. [31]. In the article, the authors merged real-world city images and 3-D rendered synthetic images, and also tried images generated with GANs. The authors prepared training sets with varying real-to-synthetic ratios (0%, 5%, 10%, 20%, 50%, 100%) and varying total dataset sizes, sampling each proportion five times. A YOLOv3 model [35] pretrained on the COCO

dataset [20] with two different methods, i.e., pretraining with synthetic data and then fine-tuning with real data, and training directly on a mixed dataset. The authors employed mAP50 as the evaluation metric. The results showed there was no actual distinction in training strategies (fine-tuning and direct training), and the mixture of real and synthetic images showed the possibility for the reduction of real images by up to 70% without reducing the performance. The most significant gains were for underrepresented classes; however, with more dominant classes having less significant gains. The GAN-generated images performed slightly worse than the 3-D rendered images, but the network trained on them also reached the same level of performance as the model trained with just the real images.

To the best of our knowledge, and from the literature analyzed, no prior work has considered studying the ratio of real and synthetic images in neural network training for runway detection. This study, therefore, aims to fill that gap, providing a helpful guideline for future neural network-based runway detection systems. By offering an early estimation of the size and ratio of the dataset that is needed for effective training, it provides guidance on creating datasets, a task that is often time-consuming and laborious. In the long term, our findings might help advance technologies supporting single-pilot and autonomous aircraft operations.

III. Methodology

Burdorf et al. [31] describe two approaches relevant to this study: pretraining on synthetic images followed by fine-tuning with real images, and training directly using a mixed dataset comprising both image types. Since there are no significant differences, we chose to train directly with the mixed dataset, since we will have to train only once. We will use a composition of the datasets from BARS [26] for the synthetic images and a real dataset we built by extracting frames from real aircraft landing footage. BARS was chosen due to its large size, diversity of airports, and public availability. The dataset contains 10,256 images generated using the X-Plane simulator. These images were taken from 40 different airports in 15 different countries, with the majority being international airports, such as Zurich Airport, Beijing Daxing Airport, and Los Angeles International Airport. For better generalization, the dataset has images in diverse weather conditions, at different distances, and with multiple runways in sight. The BARS dataset is divided into a training set, a validation set, and a test set. Figure 1 shows a few images from the dataset. Although there are other options of datasets available such as LARD [9], we want to study the use of simulators since it is the most practical way of simulating different airports, aircraft positioning, and weather conditions.

The real image dataset was constructed from 13 Embraer flight videos captured at various airports, encompassing different times of day and a range of weather conditions, including clear skies, rain, and fog. Since the videos are proprietary, they cannot be shared. From the footage, we took 1 frame each second of the video, which resulted in 3,470 images. Then we manually annotated the keypoints in each image as shown in Fig. 2.

Fig. 2 Example of manually annotated image (image extracted from public video*).

Fig. 1 Example of images contained in the BARS dataset, showing the different runways, weather, lighting conditions, distances, and multiple runways sight.

From all the real images, we separated 339 for the test set, and the remaining 3,131 images will be used for training. The dataset of 100% real images will use all 3,131 real images, and the dataset of 0% real images will use 3,131 images from BARS, which will be sampled equally from the different airports presented in the dataset. For training, the dataset will be split into a training set, which will update the weights, and a validation set, which will be used for early stopping, with 90% (2,818) and 10% (313) of the images, respectively.

To assess the effects of different dataset compositions, the NN models are trained on datasets starting with 100% real images. In subsequent iterations, 10% of the total images contained in the training and validation sets will be systematically replaced with synthetic images, and the model retrained at each step, continuing this process until the dataset consists entirely of synthetic images, as shown in Fig. 3. For the sake of simplicity, the different proportion datasets will be called by their proportion of real images. For example, the dataset which is composed of 80% of real images and 20% of synthetic images is called 80% dataset.

*AVIATIONHUB, Airbus A330 - Approach and Landing in Malé, Maldives (ENG sub), YouTube, 28 aug 2018, available at: `https://youtu.be/MTYB3wumsv8?si=0gIJN_je57_V7EJi`. Date of access: 01 jul 2025.

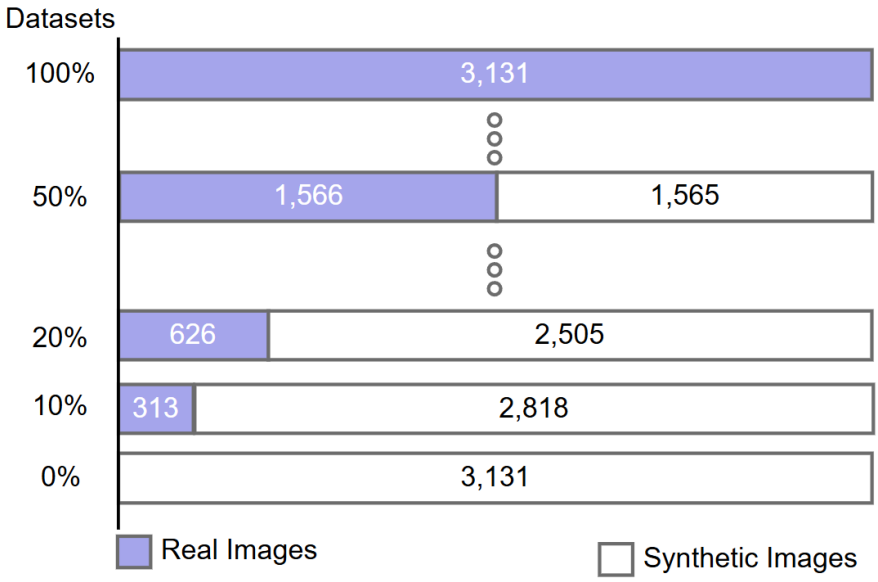


Fig. 3 Datasets' proportions of synthetic and real images in training and validation sets, then the upper bar represents a dataset with only real images, and the lower bar, a dataset with only synthetic images, while the middle ones represent the mix of the images.

Since testing every possible combination of real and synthetic images mix would be too time-consuming, we will sample each proportion (except 0% and 100%) 10 times. For example, the 30% dataset has 919 images from the 100% dataset, equally sampled by the 13 flights, and 2,212 synthetic images from the 0% dataset, equally sampled from the different airports. Then these images will be split into a training set and a validation set in a proportion of 90% and 10% respectively. Following the same sampling process, 92 images will be sampled by flight from the real images and 221 images will be sampled by airport from the synthetic images, forming the validation set, while the remaining images will form the training set. This sampling process will result in 10 different datasets for each proportion.

The objective of the sampling process is to infer the mAP of each proportion, which is a performance metric based on Object Keypoint Similarity (OKS). To obtain statistical significance of the result, showing that the obtained results do not depend on a specific sampling from the images, we will use the standard deviations similar to Ma et al. [4] and Burdorf et al. [31].

For the test set we kept the 339 images removed from the real images dataset, and we made sure those images were not mixed in the other sets. Therefore, our test is composed solely by real images which are not present in training and validation sets. This choice assumes that the operators know the runways they will operate and the training will use images from such runway, and the test set will assess the ability of the neural network to detect such runways. Therefore, this test set will be called Known Runway Test Set.

To better evaluate the generalization capability of the trained NN on unseen runways, we constructed a specific test set, which we will call Generalization Test Set [†], to evaluate the generalization capability of the trained NN when facing completely new runways. This test set is composed of 270 images from 24 different real landing footage from the web, which were sampled every 3 seconds. To avoid wrong predictions with elements from the cockpit, the videos were edited to remove them by zooming in on the area where the runway appears and cropping out the elements of the cockpit. The different features presented in the videos are: location — open field, urban landscape, and presence of bodies of water; weather — clear weather, rain, and fog; time — throughout the day until sunset; runway — single and multiple occurrences. This diversity is critical to rigorously assess the model's robustness in real-world deployment scenarios.

The task we selected for our study is keypoint detection due to the possibility of the usage of its results for other studies, such as the pose estimation done by Lima et al. [6], which would be beneficial for those who study automatic landing of aircraft using NNs. The keypoints chosen are the corners of the runway as shown in Fig. 2. To perform the task, we selected YOLOv11, as explained in Section 2.3, more specifically, the pose estimation version, which extracts the keypoints. Regarding the size of the network, the following pretrained versions are employed in the experiments:

[†]Daniel, Gustavo Hayashi. Real-Aircraft-Landing-Footage-Dataset, 2025. Github. Available at: https://github.com/Hayniel/Real-Aircraft-Landing-Footage-Dataset. Date of access: 01 nov 2025.

nano, medium, and extra large. We are using pretrained models to leverage the weights and accelerate the model training, and the different versions were chosen to evaluate whether the depth of the network might influence the need for real images. The training was performed using a workstation with an RTX 3090, Intel Core i9-11900K, and 64GB of RAM.

During training, we used the default early stopping criteria, which stops the training if there is no improvement in the total loss (sum of all loss functions for the validation set) with a patience of 100 epochs. We also employed the default hyperparameters[‡] of YOLOv11 [35], except the batch size, which we adopted YOLO's autobatch function that chooses a batch size according to the GPU memory (batch size which uses 60% of the available memory). To evaluate the model's performance, we employed the Mean Average Precision (mAP) metric. Given that the task involves extracting specific runway corners, we utilized Object Keypoint Similarity (OKS) as the underlying proximity measure, rather than the traditional Intersection over Union (IoU) utilized for bounding boxes. OKS calculates the Euclidean distance between predicted and ground truth keypoints, normalized by the object's scale. We report results using mAP50-95, which averages the precision over 10 step thresholds (from 0.50 to 0.95)[36]. This metric serves as our primary metric, as it rewards models that not only detect the runway but also pinpoint the corners with high pixel-level accuracy.

IV. Results and Discussions

First, we trained all NN with different proportions of real and synthetic images until the early stopping criteria. Figure 4 shows an example of the losses calculated during the training of the Nano version with the 30% dataset. The box loss (Fig. 4 a) measures how well the predicted bounding boxes align with the objects in the images given how far are the predictions from the ground truth, pose loss (Fig. 4 b) measures how well the model predicts keypoint locations compared to the ground truth, class loss (Fig. 4 c) measures how accurately the model predicts the class of detected objects, and distribution focal loss (Fig. 4 d) is the loss function focused on the performance on hard-to-detect examples adding weight to more challenging instances [37]. All training had similar loss curves to the ones shown in Fig. 4 showing that the training converged.

A. Known Runway Test Set

After training the models, we evaluated their performance for different proportions of mixed datasets as shown in Fig. 5, which shows the mAP in a range of 0.90 to 1.00. The performance of the model trained with 0% is much lower than the other models, and it is not shown in the figures since showing that would hinder the visualization of the performance of the other models.

Analyzing Fig. 5, it is noticeable that the performances for different sizes of NNs are similar. Since all models sizes presented a steep decrease in performance under 10% of real images, we also tested lower proportions. However, this time only for YOLO Nano assuming Medium and Extra Large will behave similarly. The performance of these proportions is shown in Fig. 6.

In Fig. 6 we can see a smoother decrease in performance when reducing the proportion of real images. It is important to note that even low proportions of real images might result in a decent performance, assuming the real images are representative of the scenario the model will face when deployed. However, depending on the scenario, one might want the performance as close as possible to a neural network trained solely with real images. For this situation, a study of a trade-off between performance and cost is necessary, especially considering scenarios such as commercial aviation, where safety is the greatest priority. For a dataset like ours, a proportion between 10 and 30% of real images could be a good starting point to train a runway detection NN, since the performance at this range of proportions is close to the 100% dataset, and an increase in proportion does not add much performance. Consequently, the use of fewer real images can be achieved, resulting in substantial cost savings due to the challenges associated with acquiring such data. However, for a real-world deployment, this proportion might change since our dataset is limited in the variability of scenarios.

The average mAP and standard deviation of all samples, for each proportion, are shown in Table 1, which has the performance for YOLO Nano, Medium, and Extra Large, including the values of mAP that were not visible in Figs. 5. The bold entries in the table are the best performance of each NN model.

The test results show consistent results for all the different samples, as shown by the small standard deviations. We can also confirm that the difference in performance between the 30% dataset and 100% dataset for all models trained is small, being less than 1%. Note that lower proportions might be used depending on the application and the performance needed.

We also performed tests to evaluate how well the model performs using only the real data. In other words, we

[‡]ULTRALYTICS, Configuration, 2025. Available at: `docs.ultralytics.com/usage/cfg/`. Date of access: 01 jul 2025.

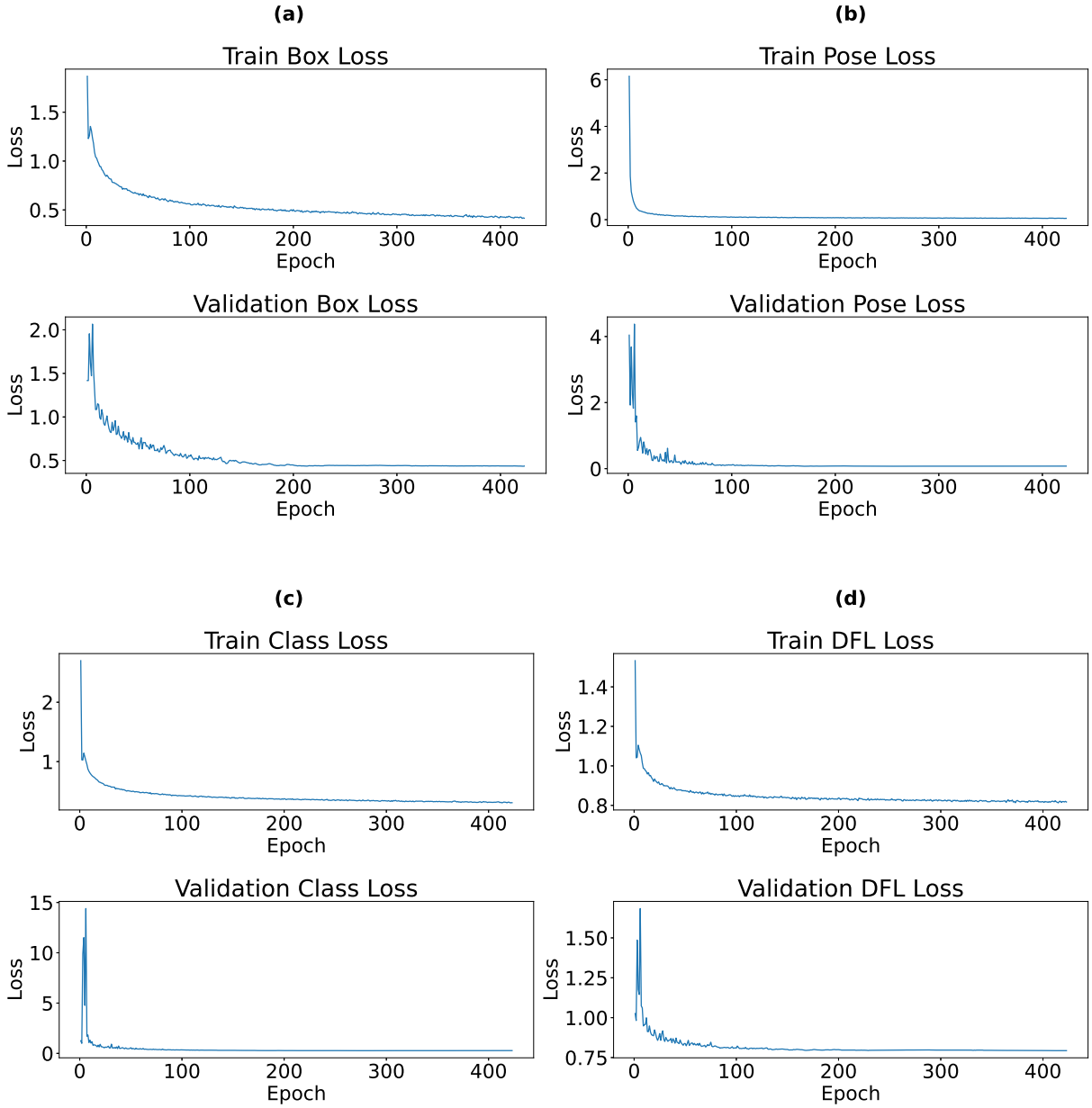
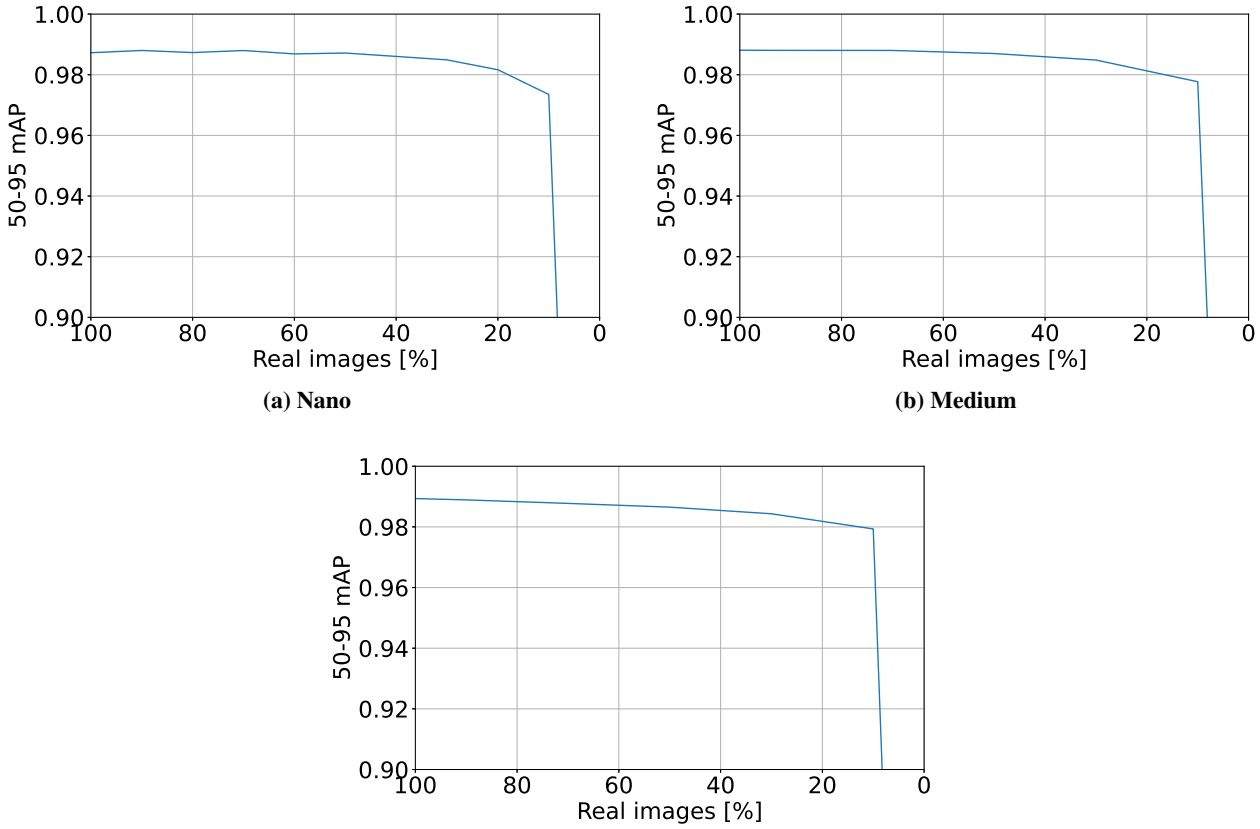


Fig. 4 Loss functions over the epochs for the validation set demonstrating convergence: (a) Box Loss and (b) Pose Loss (top row); (c) Class Loss and (d) Distribution Focal Loss (DFL) (bottom row).

removed all synthetic data from all the datasets we built, leaving only the real images. Therefore, only in Table 2, the results shown are for the real images only, with the best performance in bold. This way, we can evaluate whether the synthetic data is improving the performance of the model. For this test, we did not use all datasets; we excluded the 2%, 4%, 6%, 8%, 20%, 40%, 60%, and 80% datasets, since we already saw that fewer datasets are enough to represent the behavior of the performance.

When comparing the results from Table 1 with Table 2, we notice an increase in performance when adding the synthetic data with proportions under 30%, with significant gains under 10%, as shown in Fig. 7. Therefore, if a dataset accurately represents the deployment environment, adding synthetic images might not improve the model's performance if the dataset is already large enough to saturate it, and acquiring synthetic images would be a waste.



(a) Nano

(b) Medium

(c) Extra Large

Fig. 5 mAP 50-95 performance on the Known Runway Test Set across different model sizes. Note the similar saturation point around 10-30% real images for all architectures.

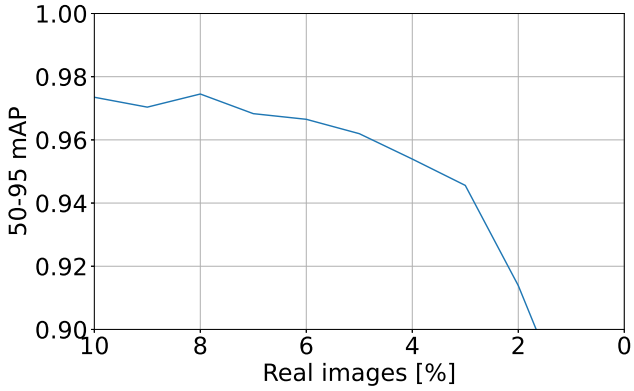


Fig. 6 Known Runway Test Set results for mAP50-95 for YOLO Nano using proportions ranging from 10% to 0%.

From Table 2, we notice that a model trained with a small amount of real images, as few as 31, might outperform a model trained solely with many synthetic images (performance of the 0% dataset of Table 1), suggesting that the quality of synthetic images is still distant from the real images, which is a domain gap. While useful, synthetic images have different properties, such as textures and lighting. Consequently, investment in real data gathering might be more beneficial for performance in a trade-off study. However, it should be noted that increasing the number of real images beyond a certain point yields diminishing returns, as the performance achieved with 939 real images is comparable to

Table 1 Comparison of mAP 50-95 (Pose) on the Known Runway Test Set across different models. Values are presented as Mean (± Std). The symbol (-) indicates proportions not evaluated for that specific model.

Real Images [%]	Nano	Medium	Extra Large
0	0.539	0.564	0.527
1	0.873 (± 0.028)	-	-
2	0.914 (± 0.018)	-	-
4	0.954 (± 0.008)	-	-
6	0.967 (± 0.008)	-	-
8	0.975 (± 0.003)	-	-
10	0.973 (± 0.003)	0.978 (± 0.002)	0.979 (± 0.002)
30	0.985 (± 0.001)	0.985 (± 0.002)	0.984 (± 0.001)
50	0.987 (± 0.001)	0.987 (± 0.001)	0.987 (± 0.001)
70	0.988 (± 0.001)	0.988 (± 0.001)	0.988 (± 0.001)
90	**0.988** (± 0.001)	0.988 (± 0.001)	0.989 (± 0.001)
100	0.987	**0.988**	**0.989**

Table 2 Performance of YOLO Nano trained solely on Real Images (no synthetic data). Values are presented as Mean (± Std). Note the rapid performance saturation with relatively small datasets.

Real Images [%]	Dataset Size (Images)	mAP 50-95 Pose
1	31	0.723 (± 0.269)
3	93	0.901 (± 0.068)
5	156	0.934 (± 0.028)
7	219	0.952 (± 0.015)
9	281	0.967 (± 0.014)
10	313	0.966 (± 0.011)
30	939	0.985 (± 0.002)
50	1,565	0.987 (± 0.001)
70	2,191	0.988 (± 0.001)
90	2,817	**0.988** (± 0.001)
100	3,131	0.987

that obtained with 3,131 real images. When comparing Tables 1 and 2 above 1,000 real images there is no significant gain in performance when adding synthetic images, suggesting that when there is enough real images, adding more synthetic images does not bring a performance improvement, meaning that training the model using only the real images in hand before mixing the dataset to check the model performance might be a good practice.

Therefore, the composition in the range of 10% to 30% of real images can be a good starting point for a dataset of around 3,000 images like ours. For larger datasets, with many synthetic images, 1,000 images might be a good starting point, since the 939 real images were enough to have a comparable performance with the 3,131 real images, although more studies are necessary, since our real dataset used only 13 flights, all during the day. Thus, for more complex datasets, these numbers might vary since our dataset still needs images in more conditions, such as snow, dawn, dusk, and other runways. Still, the total number of real images needed to achieve high levels of performance might be reasonable, without the need to have many images of each real-world condition. It is important to note that we used YOLO pre-trained on the COCO [20] dataset, and if we had used a NN trained from scratch, the results might have been different.

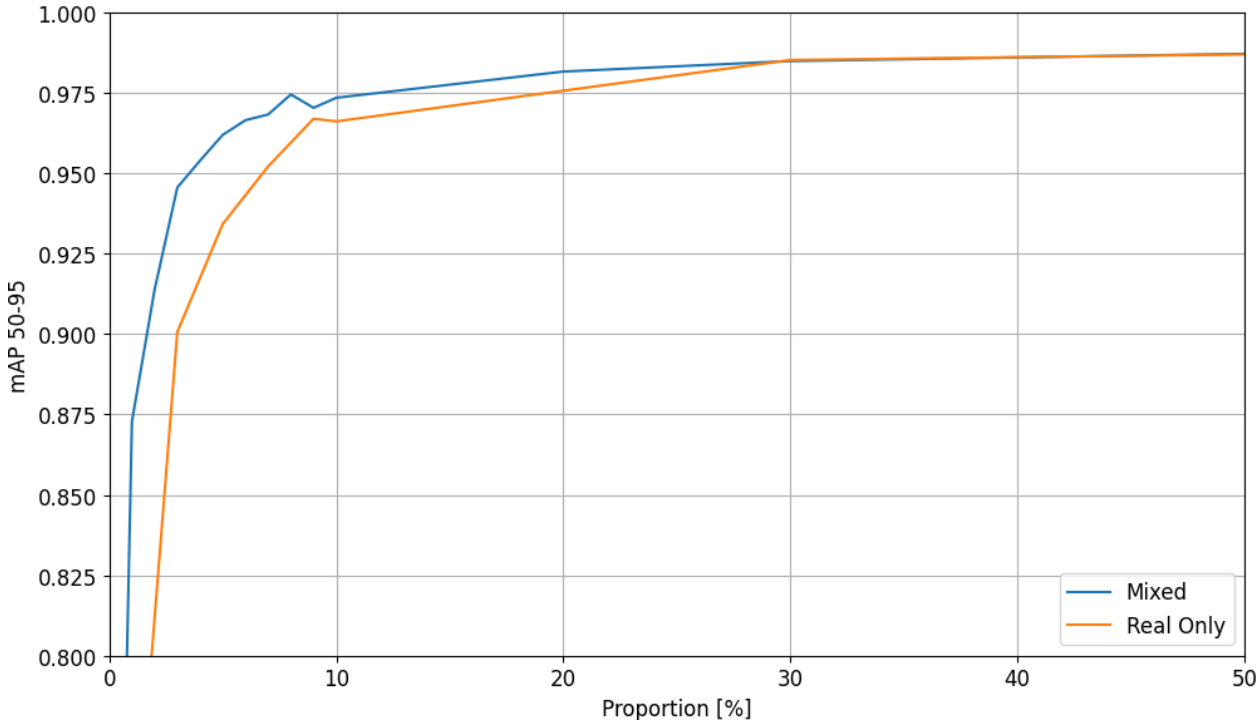


Fig. 7 Comparison of mAP between YOLO Nano trained with a mixed dataset (blue) and only with real images (orange), tested on the Known Runway Test Set. For the model trained with real images only, the proportion in the x-axis is the number of images used for training in relation to the 3,131 total real images, while for the model trained with the mixed dataset, the proportion is the number of real images in relation to the total images in the dataset.

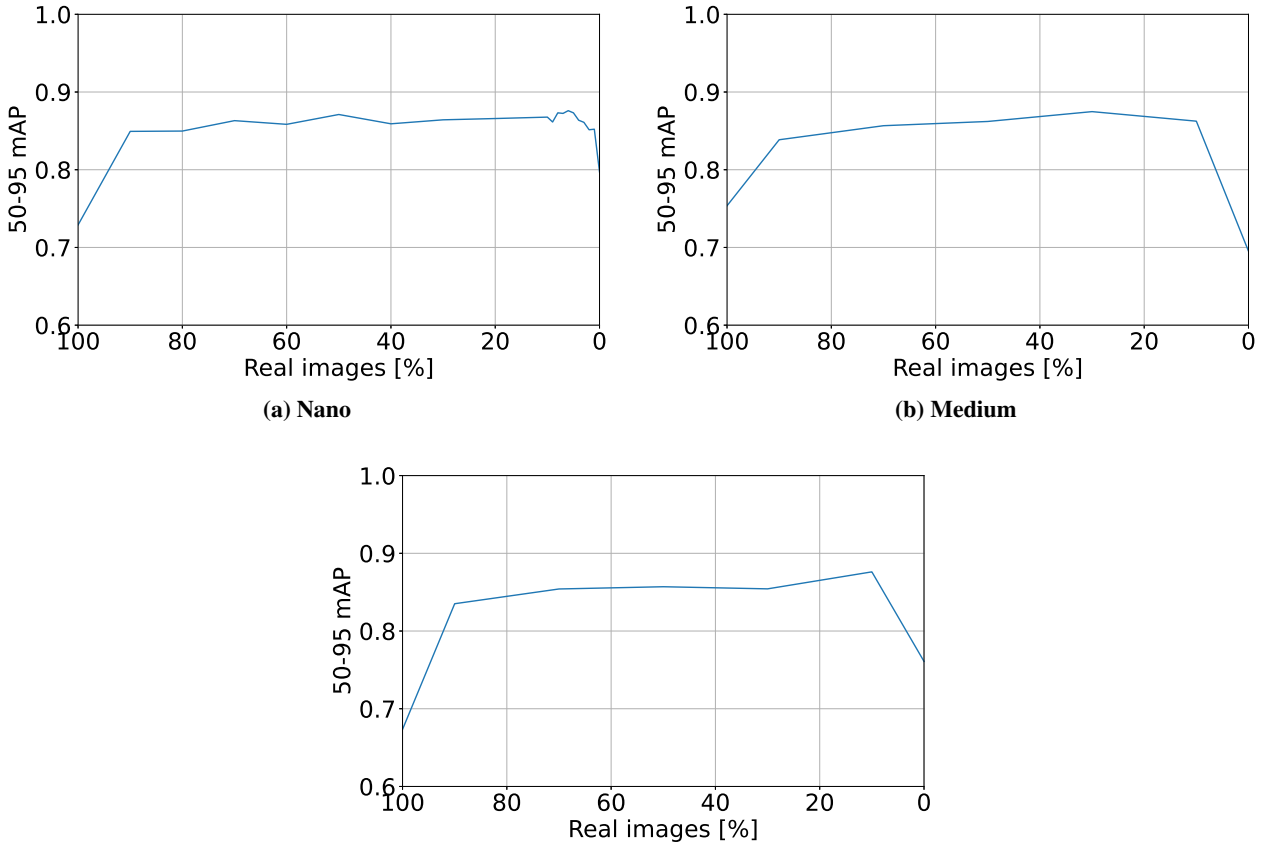
B. Generalization Test Set

Since acquiring a dataset that represents every scenario a model might face on deployment is challenging, we also tested a generalization test set, which has a greater variety of scenarios when compared to our real training set. Analogous to the Known Runway Test Set, all the sizes of NN tested presented similar behavior, as shown in Fig. 8. However, this time the models showed an increase in performance upon replacing real images with synthetic ones, until reaching a steep decrease with 0% real images. The greatest gains in performance occur in proportions between 1% and 20% real images. Our tests also show that using 100% of real images is not optimal when it comes to generalization; the synthetic images are fundamental to increasing the performance of the model by adding different scenarios to the dataset.

Table 3 shows the results in more detail with the standard deviation for each of the means taken and the best performance in bold. Again, the standard deviations are low, indicating high consistency across the random samples for each proportion, ensuring that the results are not dependent on a specific selection of images, and the tables show that the best performance is in proportions with 30% or less real images, in the case of the Nano model, 6% to 7% real images presented the best performance, reinforcing that having many real images might not translate directly to a better performance, and the fidelity to the many real scenarios should be taken into account as well, to extract the best performance possible from the real images.

Figure 9 has examples where the models trained solely with real images (0% Dataset) or synthetic images (100% Dataset) do not have the same level of performance as the model trained with a mixed dataset (10% Dataset). It is also demonstrated that the model trained exclusively on real images exhibits inferior performance compared to the model trained solely with synthetic images, as shown in the column Example 1 in Fig. 9.

Figure 10 shows how much the performance increases by adding the synthetic images to a dataset, summing up to 3,131 images. The performance gain is noticeable across the whole range of proportions, not only with a few real images. This shows how a diverse synthetic dataset can increase the performance of an NN. This situation represents cases where acquiring real data from a diverse environment is difficult, limiting the diversity of the real dataset, emphasizing the possible gains of mixing real and synthetic images when building a dataset for runway detection.



(a) Nano

(b) Medium

(c) Extra Large

Fig. 8 mAP 50-95 performance on the Known Runway Test Set across different model sizes. Note the similar saturation point around 10-30% real images for all architectures.

Table 3 Comparison of mAP 50-95 (Pose) on the Generalization Test Set across different models. Values are presented as Mean (± Std). The symbol (-) indicates proportions not evaluated for that specific model.

Real Images [%]	Nano	Medium	Extra Large
0	0.797	0.695	0.761
1	0.852 (± 0.025)	-	-
2	0.851 (± 0.013)	-	-
4	0.864 (± 0.015)	-	-
6	**0.876** (± 0.014)	-	-
8	0.873 (± 0.010)	-	-
10	0.868 (± 0.015)	0.862 (± 0.018)	**0.876** (± 0.025)
30	0.864 (± 0.018)	**0.875** (± 0.016)	0.854 (± 0.023)
50	0.871 (± 0.011)	0.862 (± 0.014)	0.857 (± 0.011)
70	0.863 (± 0.021)	0.857 (± 0.019)	0.854 (± 0.016)
90	0.849 (± 0.010)	0.839 (± 0.021)	0.835 (± 0.021)
100	0.729	0.754	0.673

Therefore, using only real or synthetic images is not optimal for building a dataset for runway detection, since by using only synthetic images, the fidelity of the simulator might influence real-life performance, and using only real images might be challenging due to the investments needed to gather images from several different environments. Using

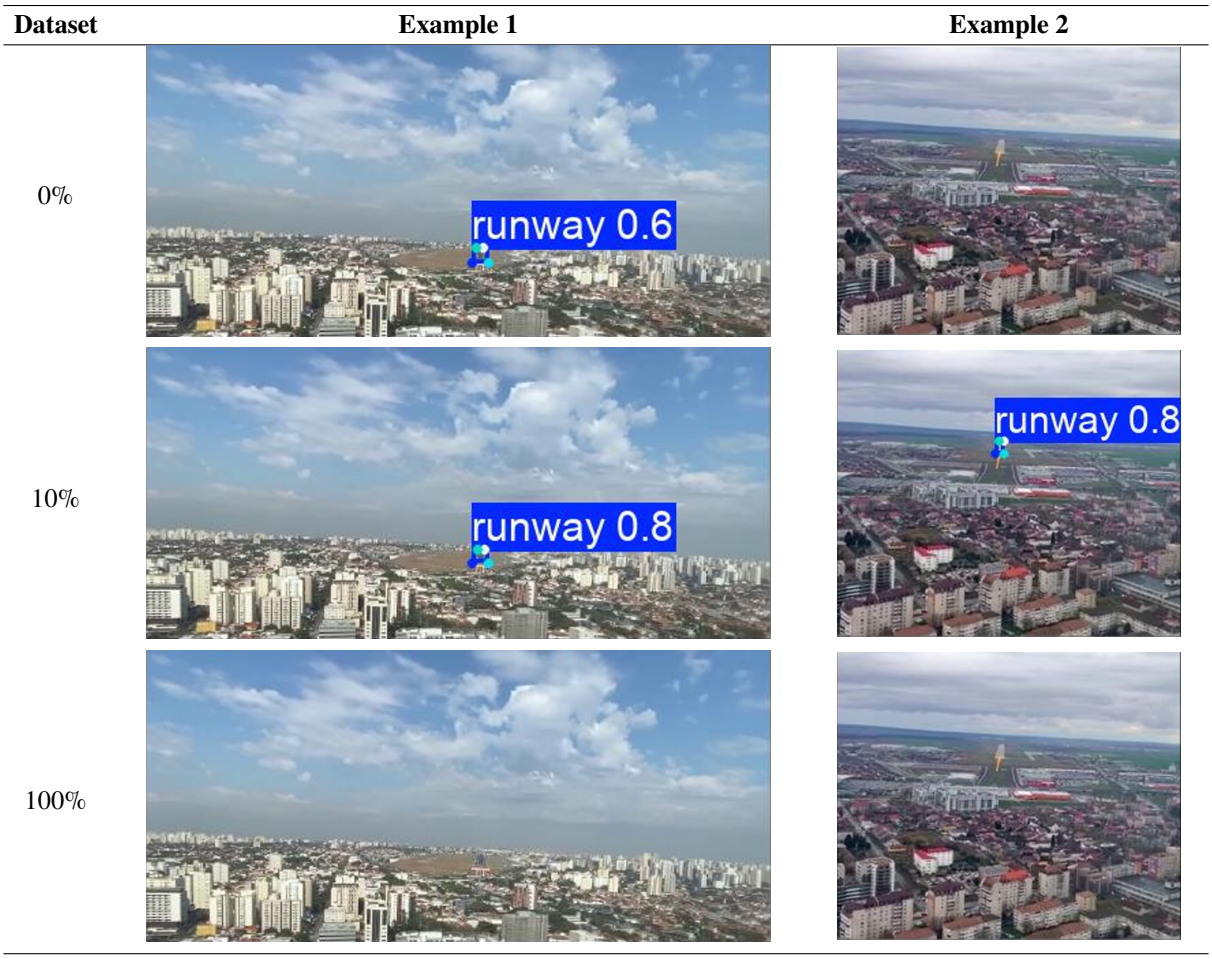


Fig. 9 Comparison between the inferences of the same model trained with different real data compositions. Each row has two predictions of a model trained with the dataset indicated in the first column. The prediction examples were taken from the generalization test set. The number on the side of each prediction is the confidence score, which is a numerical value that represents the likelihood that a specific prediction made by the model is correct [38].

a mix of synthetic and real images may be the most viable solution, given the possibility of using synthetic images to cover a wide range of scenarios, complementing the real images and achieving higher levels of performance while reducing the investments needed. By our studies, using proportions of 10%-30% of the total dataset might achieve the same level of performance as using real images alone, or even achieve greater performance depending on how well the real images represent the deployment environment.

Advancements in image quality from simulators might help to further reduce the need for real images in the near future. However, at the present time, the simulators do not provide adequate fidelity, which was evident when training the models with the known runway test set. Hence, they are not able to solve the problem of the need for a large, high-quality dataset by completely replacing the real images. Yet, current NN architectures do not need an unreasonable amount of real images to achieve high performance, meaning that future NN architectures might need even fewer real images. Our findings indicate that combining real and synthetic data is a viable approach to mitigate the challenge posed by the requirement for large datasets. From an economic standpoint, reducing the number of flights required to collect images lowers the cost of training the neural network, given the significant time and expense associated with aircraft operations and image annotation. Therefore, training on the available real images, subsequently augmenting them with synthetic data, and comparing performance, can be an effective approach to assess whether additional real images are necessary. If the model trained on the mixed dataset does not outperform the one trained solely on real images, synthetic data is unnecessary. Conversely, if performance improves, additional mixing ratios should be explored. Minor gains

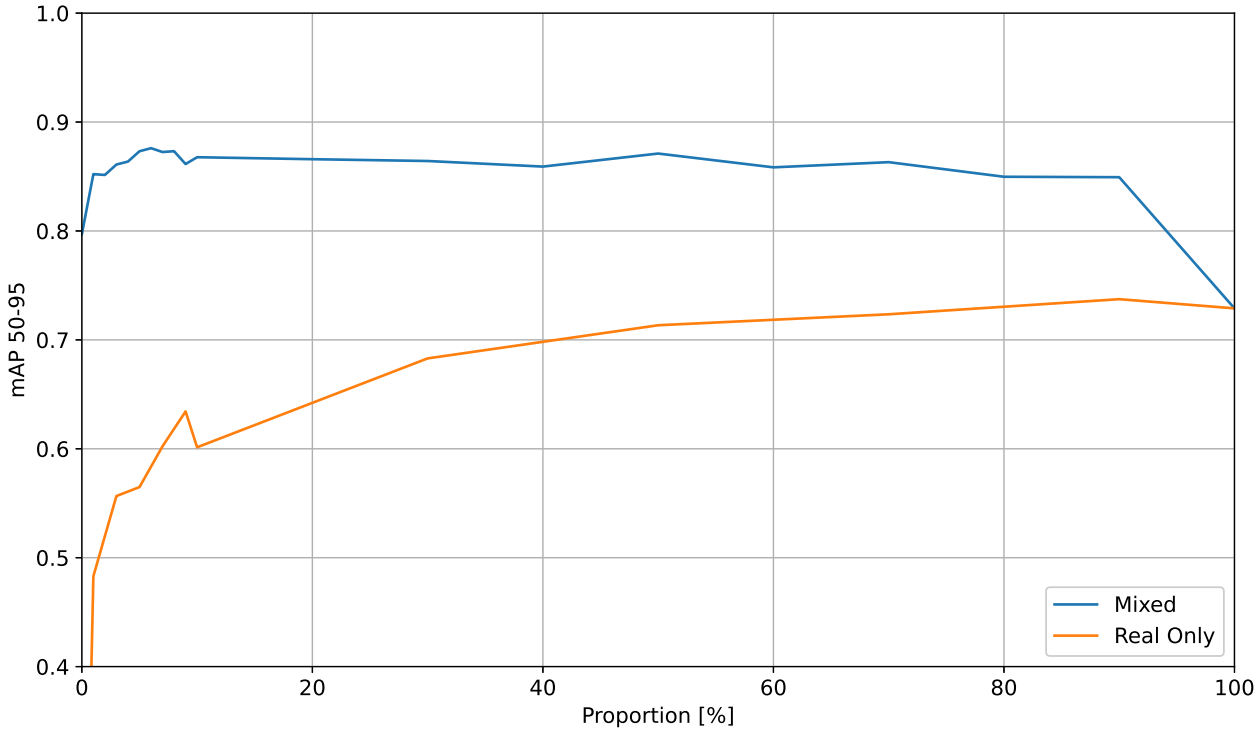


Fig. 10 Comparison of mAP between YOLO Nano trained with a mixed dataset (blue) and only with real images (orange), tested on the Generalization Test Set. For the model trained with real images only, the proportion in the x-axis is the number of images used for training in relation to the 3,131 total real images, while for the model trained with the mixed dataset, the proportion is the number of real images in relation to the total images in the dataset.

suggest that the neural network is approaching its performance ceiling, and adding more real images is unlikely to yield significant benefits. Substantial gains, however, indicate that the available real images may not adequately represent the deployment environment, implying that more real data is required. This approach enables targeted collection of only the necessary real images, thereby reducing the number of flights and associated costs.

V. Conclusion

The primary objective of this study was to conduct a case study regarding the selection and evaluation of different sizes of a neural network to analyze the performance degradation that occurs when real images are progressively replaced with synthetic images generated by simulators, given the costs and logistical challenges of gathering a large high-quality real-world dataset. We trained different sizes of YOLOv11 to perform runway corner point extraction, and we tested several proportions of mixed real-synthetic datasets, created using the BARS dataset and images extracted from real landing footage.

The key contribution of this work is the quantitative definition of a data collection guideline for runway detection tasks, in scenarios of known runways and generalization. We found that in cases where the runways on which the aircraft will land are well known, using a proportion of 10% to 30% of real images might be a good starting point, considering a dataset of 3,000 images, since this composition will have a similar performance as a dataset with 100% real images of the same size. For larger datasets, 1,000 real images may be a suitable starting point, regardless of the proportion, as our tests revealed that the difference in performance between 1,000 and 3,000 real images is minimal and adding synthetic images might not bring significant performance improvement in this specific case of well known runways, since the real images already represent well the runways and the synthetic images lack enough fidelity. Therefore, training on the available real images and then augmenting the dataset with synthetic images might be an alternative to evaluate the need for more real images in larger datasets.

For cases involving diverse runway scenarios, the usage of synthetic images proves beneficial, as acquiring a

sufficiently diverse real dataset is often challenging. A diverse synthetic dataset offers a feasible alternative and serves as a vital complement to real images. In our studies, a proportion of around 1% to 20% of real images yielded the best results in generalization tests. Unlike the known runway test set, incorporating a large number of real images actually reduced model performance. This occurred because the gathered real images did not fully represent the necessary variety of scenarios, revealing a domain gap in our dataset. The synthetic images, therefore, improved performance by effectively bridging this gap. For more complex datasets with broader scenarios, the optimal amount of real images might vary, necessitating further testing. Finally, since we utilized a NN model pre-trained on the COCO dataset [20], training from scratch could lead to different outcomes.

We hope that the results of our study help future studies for runway detection with NN architectures by providing a starting point for the dataset, as gathering real images can be time-consuming and resource-intensive, potentially diverting resources from other areas and accelerating the employment of NN models in new applications. We also hope our study encourages similar studies, as the need for a large quality dataset to train NNs effectively is not exclusive to key point extraction but is relevant to deep learning in general and other datasets beyond runway ones.

For future studies, a trade-off between investment in real images and performance improvement could be made by showing how much it costs to improve performance. Another possibility would be the implementation of a fully autonomous landing system, which needs to integrate several advanced technologies to reliably perform all the procedures needed: e.g., a control loop for precise navigation and landing on a designated runway, a data fusion algorithm to combine inputs from multiple sensors, and ML model(s) to estimate the aircraft's position regarding the runway. Robustness improvements in the ML model could also be done by adding to visible spectrum imagery different spectra or technologies, e.g., infrared, LIDAR. Another possibility to enhance landing safety could be the use of an object detection system capable of identifying obstructions on the runway, which could help to trigger go-around maneuvers in the event of a blocked or unsafe landing path. Additionally, future work could also investigate whether the ratio of real and synthetic data changes after performing a hyperparameter optimization in the model for each mixed dataset, since it was not explored in this study, as it was based only on the YOLO default hyperparameters in order to ensure a controlled comparison.

Acknowledgments

The authors would like to acknowledge the good cooperation, contributions, and support of Embraer through the PEE program. The authors also acknowledge the help of Grammarly, which is an AI-powered writing assistant, in correcting the text, and Gemini, Google AI assistant, in tables and figures formatting. Marcos R. O. A. Maximo is partially funded by CNPq – National Research Council of Brazil through the grant 307525/2022-8.

References

[1] Sapkota, B., Popescu, S., Rajan, N., Leon, R., Reberg-Horton, s., Mirsky, S., and Bagavathiannan, M., "Use of synthetic images for training a deep learning model for weed detection and biomass estimation in cotton," *Scientific Reports*, Vol. 12, 2022, p. 19580. https://doi.org/10.1038/s41598-022-23399-z.

[2] Tsapparellas, K., Jelev, N., Waters, J., Brunswicker, S., and Mihaylova, L., "Vision-based Runway Detection and Landing for Unmanned Aerial Vehicle Enhanced Autonomy," *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*, Institute of Electrical and Electronics Engineers (IEEE), New Jersey, USA, 2023, pp. 239–246. https://doi.org/10.1109/ICMA57826.2023.10215523.

[3] Agnihotri, A., Saraf, P., and Bapnad, K. R., "A Convolutional Neural Network Approach Towards Self-Driving Cars," *2019 IEEE 16th India Council International Conference (INDICON)*, 2019, pp. 1–4. https://doi.org/10.1109/INDICON47234.2019.9030307.

[4] Ma, B., Wei, X., Liu, C., Ban, X., Huang, H., Wang, H., Xue, W., Wu, S., Gao, M., Shen, Q., Mukeshimana, M., Abuassba, A. O., Shen, H., and Su, Y., "Data augmentation in microscopic images for material data mining," *npj Computational Materials*, Vol. 6, No. 1, 2020, p. 125. https://doi.org/10.1038/s41524-020-00392-6, URL https://doi.org/10.1038/s41524-020-00392-6.

[5] Kim, A., Lee, K., Lee, S., Song, J., Kwon, S., and Chung, S., "Synthetic Data and Computer-Vision-Based Automated Quality Inspection System for Reused Scaffolding," *Applied Sciences*, Vol. 12, No. 19, 2022. https://doi.org/10.3390/app121910097, URL https://www.mdpi.com/2076-3417/12/19/10097.

[6] Lima, V. A., Maximo, M. R. O. d. A., and Sêcco, N. R., "Airport Runway Detection Using Convolutional Neural Networks," Master of engineering, Instituto Tecnológico de Aeronáutica, São José dos Campos, 2022.

[7] Weng, X., and Guo, B., "A method of airport runway dataset construction for the visual detection algorithm," *Journal of Physics: Conference Series*, Vol. 2435, No. 1, 2023, p. 012016. https://doi.org/10.1088/1742-6596/2435/1/012016, URL https://dx.doi.org/10.1088/1742-6596/2435/1/012016.

[8] Shah, S., Dey, D., Lovett, C., and Kapoor, A., "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," *Field and Service Robotics*, edited by M. Hutter and R. Siegwart, Springer International Publishing, Cham, 2018, pp. 621–635.

[9] Ducoffe, M., Carrere, M., Féliers, L., Gauffriau, A., Mussot, V., Pagetti, C., and Sammour, T., "LARD - Landing Approach Runway Detection - Dataset for Vision Based Landing," , Apr. 2023. URL https://hal.science/hal-04056760, working paper or preprint.

[10] Boeing, "Statistical Summary of Commercial Jet Airplane Accidents: Worldwide Operations | 1959-2024," , 2025. URL <https://www.boeing.com/content/dam/boeing/boeingdotcom/company/about_bca/pdf/statsum.pdf>.Dateofaccess:18nov2025.

[11] Shappell, S., Detwiler, C., Holcomb, K., Hackworth, C., Boquet, A., and Wiegmann, D., "Human Error and Commercial Aviation Accidents: A Comprehensive, Fine-Grained Analysis Using HFACS," Technical Report DOT/FAA/AM-06/18, Federal Aviation Administration, Office of Aerospace Medicine, 2006. URL https://commons.erau.edu/publication/1218.

[12] Öktemer, E., and Gultekin, E., "Operational usage and importance of instrument landing system (ILS)," *International Journal of Aeronautics and Astronautics*, Vol. 2, 2024, pp. 18–21.

[13] Kügler, M. E., Mumm, N. C., Holzapfel, F., Schwithal, A., and Angermann, M., "Vision-Augmented Automatic Landing of a General Aviation Fly-by-Wire Demonstrator," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019. https://doi.org/10.2514/6.2019-1641, URL https://arc.aiaa.org/doi/abs/10.2514/6.2019-1641.

[14] Akbar, J., Shahzad, M., Malik, M. I., Ul-Hasan, A., and Shafait, F., "Runway Detection and Localization in Aerial Images using Deep Learning," *2019 Digital Image Computing: Techniques and Applications (DICTA)*, Institute of Electrical and Electronics Engineers (IEEE), New Jersey, USA, 2019, pp. 1–8. https://doi.org/10.1109/DICTA47822.2019.8945889.

[15] Hough, P. V. C., "Machine Analysis of Bubble Chamber Pictures," *Conf. Proc. C*, Vol. 590914, 1959, pp. 554–558.

[16] Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., and Randall, G., "LSD: A Fast Line Segment Detector with a False Detection Control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 4, 2010, pp. 722–732. https://doi.org/10.1109/TPAMI.2008.300.

[17] He, K., Gkioxari, G., Dollár, P., and Girshick, R., "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 42, No. 2, 2020, pp. 386–397. https://doi.org/10.1109/TPAMI.2018.2844175.

[18] Khelifi, A., Gemici, M., Carannante, G., Johnson, C. C., and Bouaynaya, N. C., "A Deep Learning Approach For Airport Runway Detection and Localization From Satellite Imagery," *2023 IEEE Symposium on Computers and Communications (ISCC)*, IEEE Computer Society, Los Alamitos, CA, USA, 2023, pp. 1066–1069. https://doi.org/10.1109/ISCC58397.2023.10217868, URL https://doi.ieeecomputersociety.org/10.1109/ISCC58397.2023.10217868.

[19] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems*, Vol. 28, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[20] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., "Microsoft COCO: Common Objects in Context," *Computer Vision – ECCV 2014*, edited by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Springer International Publishing, Cham, 2014, pp. 740–755.

[21] Yang, L., Wu, J., Li, H., Liu, C., and Wei, S., "Real-Time Runway Detection Using Dual-Modal Fusion of Visible and Infrared Data," *Remote Sensing*, Vol. 17, No. 4, 2025. https://doi.org/10.3390/rs17040669, URL https://www.mdpi.com/2072-4292/17/4/669.

[22] Bor, N., Pervan Akman, N., and Berkol, A., "AeroRunway: Diverse Weather and Time of Day Aerial Dataset for Autonomous Landing Training," *Journal of Advanced Research in Natural and Applied Sciences*, Vol. 10, 2024, pp. 735–746. https://doi.org/10.28979/jarnas.1500916.

[23] Daixian, Z., "SIFT algorithm analysis and optimization," *2010 International Conference on Image Analysis and Signal Processing*, Institute of Electrical and Electronics Engineers (IEEE), New Jersey, USA, 2010, pp. 415–419. https://doi.org/10.1109/IASP.2010.5476084.

[24] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, Vol. 110, No. 3, 2008, pp. 346–359. https://doi.org/https://doi.org/10.1016/j.cviu.2007.09.014, URL https://www.sciencedirect.com/science/article/pii/S1077314207001555, similarity Matching in Computer Vision and Multimedia.

[25] Lukezic, A., Vojir, T., Zajc, L. C., Matas, J., and Kristan, M., " Discriminative Correlation Filter with Channel and Spatial Reliability ," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, Los Alamitos, CA, USA, 2017, pp. 4847–4856. https://doi.org/10.1109/CVPR.2017.515, URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.515.

[26] Chen, W., Zhang, Z., Yu, L., and Tai, Y., "BARS: a benchmark for airport runway segmentation," *Applied Intelligence*, Vol. 53, No. 17, 2023, pp. 20485–20498. https://doi.org/10.1007/s10489-023-04586-5, URL https://doi.org/10.1007/s10489-023-04586-5.

[27] Quessy, A. D., Richardson, T. S., and Hansen, M., "Vision based Semantic Runway Segmentation from Simulation with Deep Convolutional Neural Networks," *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2022. https://doi.org/10.2514/6.2022-0680, URL https://arc.aiaa.org/doi/abs/10.2514/6.2022-0680.

[28] Wang, Z., Zhao, D., and Cao, Y., "Visual Navigation Algorithm for Night Landing of Fixed-Wing Unmanned Aerial Vehicle," *Aerospace*, Vol. 9, 2022. https://doi.org/10.3390/aerospace9100615.

[29] Li, Y., Xia, Y., Zheng, G., Guo, X., and Li, Q., "YOLO-RWY: A Novel Runway Detection Model for Vision-Based Autonomous Landing of Fixed-Wing Unmanned Aerial Vehicles," *Drones*, Vol. 8, No. 10, 2024. https://doi.org/10.3390/drones8100571, URL https://www.mdpi.com/2504-446X/8/10/571.

[30] Fatima Ezzahra, K., Najat, R., and Abouchabaka, J., "Comparative Analysis of Transfer Learning-Based CNN Approaches for Recognition of Traffic Signs in Autonomous Vehicles," *E3S Web of Conferences*, Vol. 412, 2023. https://doi.org/10.1051/e3sconf/202341201096.

[31] Burdorf, S., Plum, K., and Hasenklever, D., "Reducing the Amount of Real World Data for Object Detector Training with Synthetic Data," , 2022. URL https://arxiv.org/abs/2202.00632.

[32] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, Vol. 27, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf.

[33] Lee, H., Kang, S., and Chung, K., "Robust Data Augmentation Generative Adversarial Network for Object Detection," *Sensors*, Vol. 23, No. 1, 2023. https://doi.org/10.3390/s23010157, URL https://www.mdpi.com/1424-8220/23/1/157.

[34] Bird, J. J., Faria, D. R., Ekárt, A., and Ayrosa, P. P. S., "From Simulation to Reality: CNN Transfer Learning for Scene Classification," *2020 IEEE 10th International Conference on Intelligent Systems (IS)*, Institute of Electrical and Electronics Engineers (IEEE), New Jersey, USA, 2020, pp. 619–625. https://doi.org/10.1109/IS48319.2020.9199968.

[35] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., " You Only Look Once: Unified, Real-Time Object Detection ," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, Los Alamitos, CA, USA, 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91, URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.91.

[36] COCO, "COCO main website," , 2025. URL <https://cocodataset.org/#keypoints-eval>.Dateofaccess:01dec2025.

[37] Torres, J., "What is DFL loss in yolov8?" Available at: https://yolov8.org/what-is-dfl-loss-in-yolov8/#:~:text=Distributed%20Focal%20Loss%20(DFL)%20is,easier%20to%20learn%20from%20them., 2024. Date of access: 12 sep 2025.

[38] Ultralytics, "Glossary - Confidence," Available at: https://www.ultralytics.com/glossary/confidence, 2025. Date of access: 1 dec 2025.