Deutscher Luft- und Raumfahrtkongress 2023
DocumentID: 610072

# NEURAL NETWORK ENSEMBLES FOR SAFETY-CRITICAL OBJECT DETECTION FUNCTIONS IN AEROSPACE

J. Sprockhoff*, U. Durak*
* German Aerospace Center, Institute of Flight Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany

**Abstract**

This paper discusses the benefits of using ensembles of neural networks for safety-critical tasks in an aircraft. The advantages of increased performance, uncertainty assessment of predictions and redundancy are highlighted. This is done using the example of an AI-based system to detect other aircraft. The system uses a stereo vision approach with two cameras to determine the distance to other aircraft. It is shown how by averaging predictions over an ensemble of five object detectors the detection rate and the accuracy of distance predictions can be improved over a single neural network.

**Keywords**
Artificial Intelligence; Neural Networks; Ensemble Learning; Computer Vision; Object Detection

## 1. INTRODUCTION

Fully autonomous flights are a long-standing goal of the aviation industry. In recent years, the significant progress in Artificial Intelligence (AI) has brought us one step closer to this goal. For example, Machine Learning (ML) algorithms utilizing deep neural networks have brought significant advances to the field of computer vision. This becomes particularly evident in tasks like object classification and object detection [1], which are essential for enabling autonomous mobility. However, to safely integrate AI into newly developed aircraft systems, a variety of new AI-related challenges must be addressed, such as constitution of training data and securing the learning process [2]. Current standards for aircraft software development, such as ED-12C/DO-178C [3], do not address these challenges and therefore cannot be used to develop AI-based systems. Therefore, the creation of new standards adapted to AI development is necessary. Amongst others, the European Union Aviation Safety Agency (EASA) is currently working towards the certification of ML-based systems for safety-critical tasks in aerospace. As a deliverable of EASA's AI roadmap [4] a concept paper titled *"First usable guidance for Level 1 & 2 machine learning applications"* was published in early 2023 [5]. This guidance document contains a set of anticipated objectives that must be fulfilled for certification. One objective stated in the guideline is to ensure that ML models can effectively generalize on operational input data. Generalization refers to the ability of an ML component to perform its task for inputs that were not previously encountered in the training data. Currently, however, the generalization ability and accuracy of single neural networks are far below the performance required to be reliably used for safety-critical tasks in aviation. This holds especially true for perception tasks like object classification and object detection, where the vast input space of images cannot be fully covered by the available training data. Another objective from the EASA guideline is the provision of indications on uncertainty and reliability by AI-based components regarding their outputs [5]. In order to be able to prevent hazardous situations introduced for example by false detections, trustworthy uncertainty assessment is indispensable for each prediction. Consequently, improved performance and the ability to estimate own uncertainty are necessary prerequisites for employing ML in safety-critical systems.

A common design pattern for systems based on ML are so-called ensembles [6]. The concept behind ensembles is to combine the predictions of multiple ML models to attain better predictions than what a single model could achieve, by means of "collective intelligence". It was firstly shown in [7] that creating ensembles of neural networks can improve the generalization ability of the system. Since then, neural network ensembles have been successfully applied in various fields such as healthcare [8], finance [9], and meteorology [10]. To use ensembles effectively, it is important to have a high degree of diversity between the individual models [11]. Diversity in ensembles can, for example, mitigate the consequences of overfitting [12]. Overfitting refers to a common problem in ML where models are to strongly adapted to the training data and therefore perform poorly when faced with unseen data, i.e. they generalize poorly. The usage of neural network ensembles for flight systems could have at least the following three benefits: First, ensemble techniques have proven to be capable of achieving improved performance when compared to single neural networks. Second, ensembles can provide a measure of uncertainty for each prediction. Third, neural network ensembles increase the dependability and robustness of the function.

doi: 10.25967/610072

In this paper the application of neural network ensembles is analyzed for an advanced air mobility use case. An ensemble of object detectors based on Yolov7 [13] evaluates images provided by two cameras to estimate the distance to an aircraft in order to identify a potential hazard, i.e. a shortfall below the permissible minimum distance. The distance-estimation system is analyzed on a set of test images created using the open-source flight simulator FlightGear [14]. The distance predictions are analyzed for their accuracy and uncertainty.

This paper is organized as follows: In Section 2 background information is provided about object detection, ensembles and uncertainty in ML. The following Section 3 describes the ensemble-based distance-estimation system and provides an explanation of the test experiment. Section 4 displays the test results regarding accuracy and uncertainty. Section 5 is discussing the results as well as further advantages and challenges of ensembles in regard to redundancy before the paper is concluded by Section 6 which provides a summary and an outlook on further research.

## 2. BACKGROUND

### 2.1. Object Detection

Compared to classical image classification, object detection algorithms not only identify the classes of objects present in the image, but also estimate their positions in pixels. The positions are represented by so-called bounding-boxes that fit tightly around each identified object (see Figure 1). This additional spatial information can be used, for example, to track the movement of identified objects. When using object detection there are two aspects to consider; firstly, that the network correctly identifies and classifies the object, and secondly, that size and position of the bounding box adequately represent the actual object. During training and testing, a detection is considered correct (true positive) only if the classification of the object is correct and the Intersection over Union (IoU) between the detected box and the ground truth box which comes annotated with the data is above a certain threshold. The IoU between two bounding boxes A and B is defined as follows:

$$\text{IoU(A,B)} = \frac{A \cap B}{A \cup B} \qquad = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

This paper encourages the usage of the IoU value of all bounding boxes from the different ensemble members that belong to a particular object as described in Section 4. In object detection two types of errors are usually distinguished: False positives, i.e., detected objects that are not actually present, and false negatives, i.e., actual objects that have been missed by the detector. Furthermore, neural networks are often evaluated using two metrics: precision and recall. Precision is the number of true positives divided by all positives, i.e., how many detections made by the model are actually existent objects. Recall is the number of true positives divided by the total number of ground truth objects, i.e., how many of the existing objects were correctly detected. Depending on the application, a high precision or a high recall may be more desirable.

In recent years, a variety of object detectors have been proposed. Object detectors are commonly categorized into two classes: two-stage and one-stage detectors [15]. Two-stage detectors first propose a set of regions in an image where objects might be present. In the second stage, these regions are analyzed and bounding boxes are proposed. With one-stage detectors the region proposal stage is omitted and bounding boxes are predicted directly. Two-stage detectors have slightly higher accuracy, but they are also slower. Therefore, one-stage detectors are often preferred when it comes to applications that require real-time detections. One-Stage detectors based on the *You only look once* (Yolo) algorithm are among the most popular. Our distance-estimation system uses the Yolov7 object detector [13] to identify aircraft and generate bounding boxes.

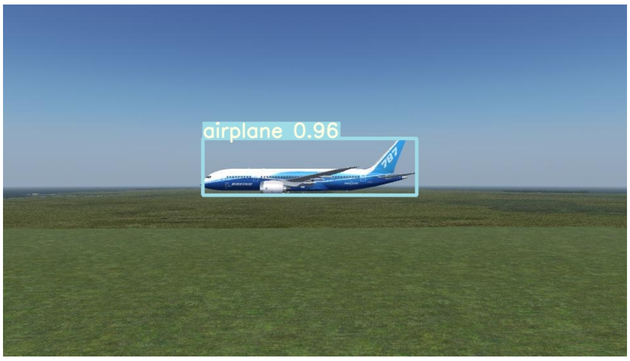

**FIG 1. Detection of an object of the class "airplane" represented by a bounding box.**

### 2.2. Ensemble Learning

The concept of ensembles in ML showed first major progress in the early 1990s and has since seen a steady increase in relevance in research [12]. Today, there are many different approaches to implementing neural network ensembles. One of the most well-known approaches to ensemble learning is the bootstrap aggregation method, also known as bagging. The concept of bagging was originally introduced in 1996 [16]. The idea of bagging is to train multiple models on different data sets sampled from an original data set and aggregate the individual prediction results. For the creation of the different training data sets the bootstrapping strategy is used. Bootstrapping is a process in which individual training samples are randomly selected from the original data set to create a new data set. Each individual data sample can be selected again for the same bootstrap data set. This means that individual samples from the original data set may appear more than once or not at all in a new data set. Usually each bootstrap data set contains the same amount of samples as the original data set. Figure 2 visualizes the approach. Each sampled data set is used to train an ML model. During inference, the predictions of all models are aggregated to obtain a more robust prediction. Depending on the problem type (regression or classification), predictions are aggregated by averaging or voting. It is possible to weight predictions differently or to filter out individual predictions. In addition, different voting strategies can be applied, depending on whether precision or recall is more important for the particular application. For an object detection system, high precision would mean that more detections would belong to actual objects and that we would receive fewer false alarms. However, it is also more likely that the system will miss actual objects which can lead to dangerous situations. On the other hand, a high recall ensures that more objects are detected, but at the cost of more false alarms. The aggregation leads to less variance in the predictions and therefore improves the performance of the system. Examples for other common ensemble methods are boosting [17] and stacking [18]. In the experiment described in this paper, the bagging approach was used to create an ensemble for our system and obtain distance predictions.

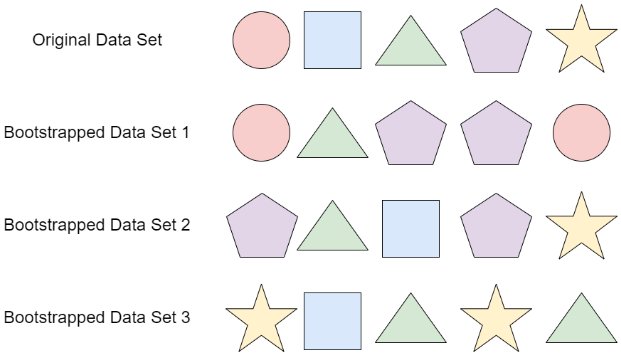

**FIG 2. Example for bootstrapping data sets. From a small data set of five samples three data sets were created by applying bootstrapping.**

### 2.3. Uncertainty

For an AI-based system to be sufficiently trustworthy to be used for a safety-critical task, it must be able to express how certain a prediction is. If the uncertainty is high, it is then possible to switch to a non-AI component. Modelling uncertainty in AI is a highly active research topic [19]. In ML, a general distinction is made between two types of uncertainty: aleatory uncertainty and epistemic uncertainty [20]. Aleatoric uncertainty is the inherent uncertainty in the data and cannot be reduced by adding more data to the training. Aleatoric uncertainty can be further divided into homoscedastic and heteroscedastic uncertainty. Homoscedastic uncertainty is constant and independent of the input data. Heteroscedastic uncertainty, on the other hand, depends on the input and can therefore be different for each sample. In object detection aleatoric uncertainty can occur, for example, in the form of image noise or motion blur, but also due to occlusion or lighting. On the other hand, epistemic uncertainty describes the uncertainty of the model. Epistemic uncertainty can arise, for example, from overfitting or an insufficient amount of training data. Unlike aleatoric uncertainty, epistemic uncertainty can be improved by adding new training data. There are different techniques to model the different types of uncertainty. A popular approach to estimate uncertainty are Bayesian Neural Networks (BNN). In BNNs weights are represented by distributions instead of fixed scalars. On each pass through the network a random sample value is picked from each weight distribution, leading to different outputs. The distribution of results helps to estimate uncertainty and reduce overfitting predictions. The posterior distribution of a BNN represents the epistemic uncertainty of the model [20]. However, calculation of the posterior distribution is not possible in general [21]. For this reasons techniques to approximate BNNs were created. In [22] it was shown that neural network ensembles can be used as an alternative to BNNs to estimate uncertainty. The authors see the advantage of using ensembles for uncertainty estimation in the ease of implementation and strong performance compared to other methods. In our experiment, the use of an ensemble also allows us to estimate the uncertainty for each prediction. The total number of positives, the distribution of the result values, and the IoU between the bounding boxes are considered together to assess uncertainty.

## 3. APPROACH

To investigate the utility of neural network ensembles for a potential use case in aviation, experiments were conducted with an exemplar distance-estimation system. The system is based on object detection and stereo vision with two cameras. The usage of two individual cameras positioned at a fixed distance from each other facing in the same direction makes it possible to calculate the distance to an object. The two cameras both simultaneously take images from the same scene. The Yolov7 object detector [13] is then used to generate bounding boxes around objects on both images. The disparity between the centers of corresponding bounding boxes in both images is used to calculate the distance to that object using the formulas from [23]. In the experiments, the distance between the cameras was set to 10 meters, the field of view of the cameras was 73.6°, and the image width was 960 pixels. It should be noted that with this method of distance calculation, the distance increases exponentially, inversely to the pixel disparity between the bounding boxes (see Figure 3). This means that for distant objects, where pixel disparity between bounding boxes is low, the number of possible distance prediction results is small and the calculations become less accurate. Increasing the distance between the two cameras or increasing the number of pixels per degree of view can improve the applicability of the system at longer distances. However, care must always be taken to ensure that objects are fully visible on both camera images, otherwise false results will be obtained due to incorrect center points of the bounding boxes. The system is tested in a virtual environment provided by the open source flight simulator FlightGear [14].

FlightGear offers the possibility to easily create own scenarios, so that models of other aircraft can be placed at any desired position. Furthermore, it has a large li-

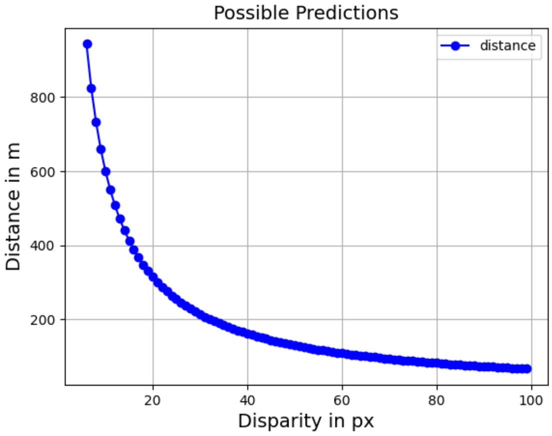

**FIG 3. This Graph displays the relationship between the calculated distance and the disparity in pixels between the two images for the parameters used in the experiment: Distance between the two cameras of 10 meters, field of view of the cameras of 73.6° and image width of 960 pixels.**

| # Samples | Distance in m | Perspective |
|---|---|---|
| 12 | 82 | Rear |
| 24 | 111 | Left/Right |
| 12 | 112 | Front |
| 12 | 131 | Rear |
| 12 | 161 | Front |
| 12 | 180 | Rear |
| 12 | 210 | Front |
| 24 | 222 | Left/Right |
| 12 | 228 | Rear |
| 12 | 258 | Front |
| 12 | 277 | Rear |
| 12 | 307 | Front |
| 24 | 333 | Left/Right |
| 24 | 444 | Left/Right |
| 24 | 556 | Left/Right |

**TAB 1. Overview of samples**

brary of aircraft models available to use and also provides different environmental settings. To evaluate the performance of our ensemble, a set of test data scenes was created using FlightGear. For this, analogous to the distance-estimation system, two cameras were defined in FlightGear with the same parameters to capture images of predefined scenes. All scenes consist of a single aircraft viewed from different angles and distances under different environmental conditions. In the scenes three models of different-sized aircraft were used (373-300, 787-8 and A340-600). The images of the aircraft models were taken from four different angles, each differing by 90 degrees, i.e., from the front, from the rear, from the left, and from the right, and from five different distances for each angle. This was done under three different daytime settings as well as in a winter setting (see Figure 4). The result is a test set consisting of 240 annotated pairs of images (5 distances * 4 angles * 4 day times * 3 aircraft models). Table 1 provides an overview of the distances used in the data set. Each sample from the test data set consists of the two camera images annotated with the distance to the object as ground truth. The ground truth distance was calculated from the earth coordinates of the aircraft model and the distance-estimation system. It should be noted that the test cases are purely designed to demonstrate the advantage and not to replicate a real-world scenario. The authors are well aware that for real-world application of such a system the necessary operational distance would have to be identified and parameters and cameras must be chosen accordingly.
To create the different object detection models, we used the bootstrapping method as the first step of bagging as described in Section 2. The base data set for training consisted of 4500 images from the OpenImages data set [24]. All images used were labelled as "Airplane" and without the attributes "Truncated", "GroupOf", "Depiction" and "Inside". We used Yolov7 to train five models on different bootstrap data sets of 4500 training images.

All 5 models were trained for 150 epochs each with a batch size of 24 and an image size of 640x640. To assess the performance of the ensemble, each test sample was evaluated by each of the five object detection models, such that 5 distance estimates per test sample were available in case none of the detectors missed the detection. For comparison, another model was trained on the entire base training data set without applying any ensemble techniques. The same test data set was evaluated by this single network to obtain results that can be used for comparison between the ensemble and a single neural network. The confidence-threshold for yolov7 was set to 0.7 and the IoU-threshold to 0.7 for all tests.

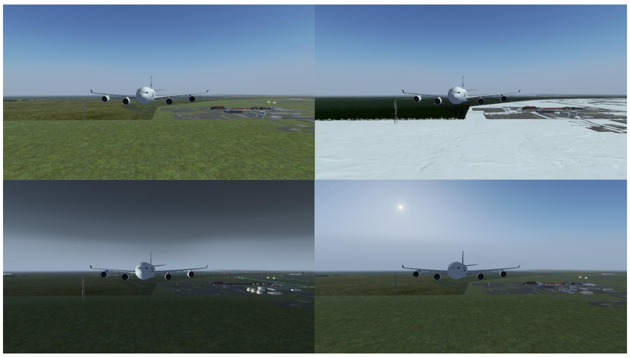

**FIG 4. The four different daytime settings used in the demonstration. Noon in summer (top left), noon in winter (top right), evening in summer (bottom left) and morning in summer (bottom right)**

## 4. RESULTS

The top diagram in Figure 5 displays the distance predictions of all five ensemble members in comparison to the ground truth distance. The ability of the system to determine the distance is clearly visible. For test cases with

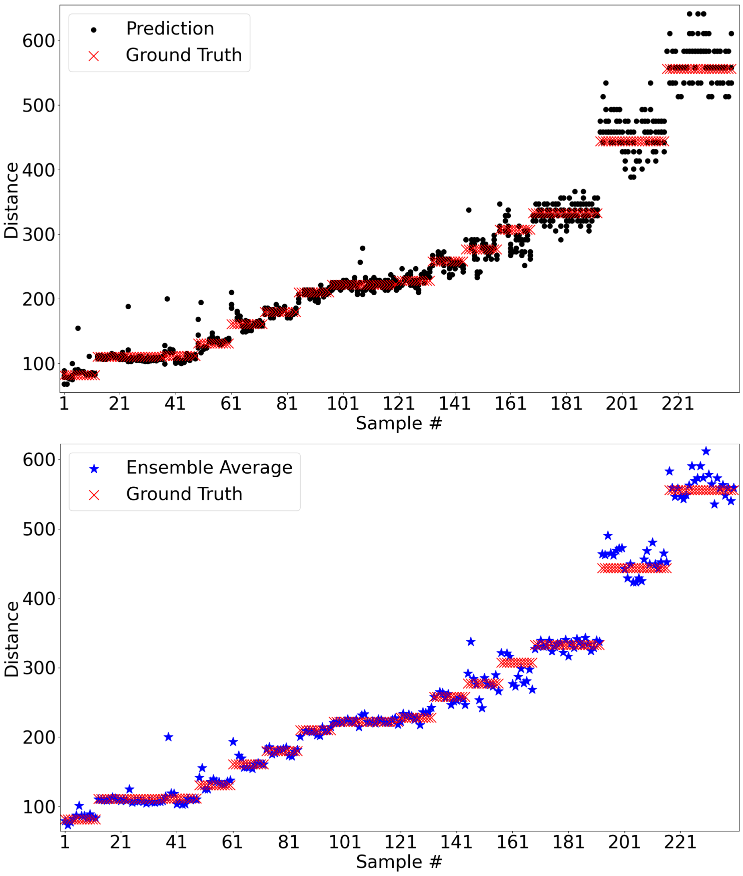

**FIG 5. Top: Predictions of the five ensemble networks versus ground truth distance for all 240 test cases. Bottom: Average value of the five ensemble members predictions versus ground truth distance for all 240 test cases**

short ground truth distance the predictions tend to be very close. For the test cases with higher ground truth distances, the deviations are significantly higher, which is not surprising due to the relation shown in the Figure 3. However, individual outlier predictions can also be found in the test cases with a short distance. The reason for this are inaccurate bounding boxes by the object detector. Overall, the predictions for the test cases where the aircraft was seen from the side were more accurate than those from the front or rear. It is also noteworthy that the predictions of the five models always tended to be slightly different. Often there were four or even five different estimates provided for one and the same test case. For large distances, this led to prediction differences of over 100 meters between ensemble members. In the bottom diagram of Figure 5 the average values of the five detectors are compared with the ground truth. If a network failed a detection, it was excluded from the average calculation of that test case. Here, the outliers were largely mitigated.

Overall, the ensemble was able to provide a prediction for 236 of the 240 test cases, meaning that at least one network in the ensemble detected the aircraft on both camera images so that a distance calculation could be performed. The four samples for which all five ensemble members could not identify the aircraft were all images of the 373-300, the smallest aircraft in the test set, taken from the front in winter conditions. This indicates an issue with the training data set which can possibly be resolved by adding appropriate images. On the other hand, the single network could not detect the aircraft in 21 test cases. This clearly shows a better recall of the ensemble, that has failed in significantly fewer tests and thus offers a higher level of safety.

However, if the average is simply taken as the result, it may happen that individual bad or wrong predictions strongly influence the result and worsen the overall prediction. Therefore, it makes sense to implement a method that detects outliers and excludes them from the calculation. Methods such as the interquartile range (IQR) method could help here. With a higher number of ensemble members the outlier detection becomes more reliable. We applied a simple approach to our ensemble prediction results, where the highest and lowest values are ignored in the average calculation if all five networks provided a prediction (averaging without extremes). If for a test sample one or more ensemble members could not give a prediction due to lack of detection, the average was calculated normally. This method resulted in an improvement, albeit slight, in predictions regarding the average deviation from ground truth (see Table 2). With more sophisticated averaging methods and more ensemble members, further improvements can be expected. Additionally, the average deviation between the median of the ensemble predictions and the ground truth was calculated. The value was slightly higher than that of the two averaging methods. In general, the difference in average deviation was rather small between the three methods considered. However, compared to the single network, all three methods showed an accuracy advantage of about 20%. These benefits become even larger if the test cases where the single network failed are omitted from the average calculation for the ensemble methods. In this case, for the "averaging w/o extremes" method, the average deviation is only 7.8479 m.

| | Avg. Deviation from Ground Truth in m |
|---|---|
| Ensemble Avg. | 8.6902 |
| Ensemble Avg. w/o Extremes | 8.6107 |
| Ensemble Median | 8.7559 |
| Single Network | 10.8181 |

**TAB 2. Average deviation of all test cases from ground truth distance for four different approaches ignoring failed detections**

The general benefit of employing ensembles for this distance determination method, reliant on object detection, becomes clearly evident upon examining the results for the most distant test cases showcased in Figure 6. Among the total of 24 test samples, the ensemble approach "averaging w/o extremes" provided predictions closer to the ground truth for 18 test samples, whereas the single network's predictions were closer for only three test samples. Since an image represents real space in pixels, the number of possible positions for object detection is limited at any time. The usage of multiple detectors allows the fusion of individual results and therefore a more fine-grained distance estimation. The results provided by the ensemble may be closer to ground truth than is possible with a single neural network.

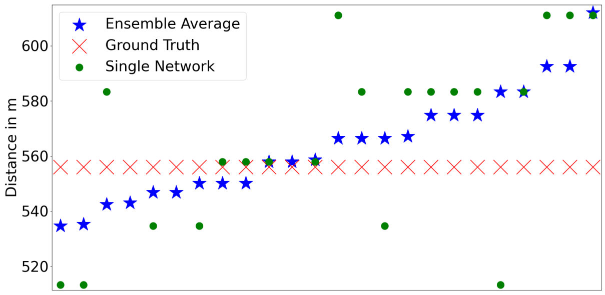

**FIG 6. Comparison of the prediction results of the ensemble and the single network for the test samples with a 556 m ground truth distance**

However, the ensemble also performed better on test samples with shorter distances, where the intervals between the possible predictions of the single network are smaller. This is exemplified, as depicted in Figure 7. For the test samples with 222 m distance the ensemble approach "average w/o extremes" achieved the closer result in 20 out of 24 cases.

To estimate the uncertainty of the prediction based on the ensemble, it is intuitive to first look at the number of detectors that identified each object. The bounding boxes of each detector can be matched and assigned to objects based on their IoU, and a voting strategy can be applied to distinguish between valid and invalid detections, similar to [25]. Furthermore, uncertainty is typically evaluated through the examination of the variance within the predictions. However, for object detection tasks it might also be useful to consider the IoU value between bounding boxes during uncertainty estimation. IoU is usually calculated to determine the similarity between two bounding boxes, but can also be calculated for multiple bounding boxes. For this purpose, the intersection area covered by all bounding boxes is divided by the area of the polygon resulting from the union of all bounding boxes. Figure 8 displays the relationship between the ground truth distance of the test sample and the IoU between all bounding boxes of the results, averaged between the left and the right image. False negative predictions by single ensemble members were ignored for the IoU calculation. Test cases with only one or zero detections are also excluded, since the IoU here is 1 or undefined, respectively, by definition. A trend can be seen that for longer distances the IoU decreases. This is mainly due to the fact that the objects appear smaller at a greater distance and thus small pixel differences in the bounding boxes are more significant.

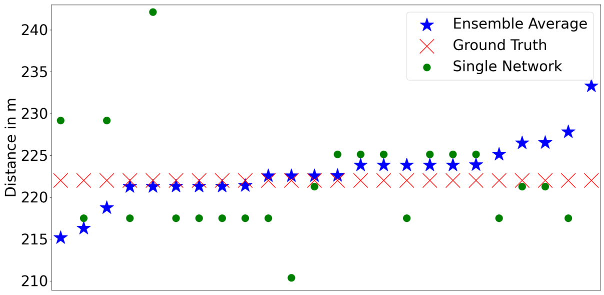

**FIG 7. Comparison of the prediction results of the ensemble and the single network for the test samples with a 222 m ground truth distance**

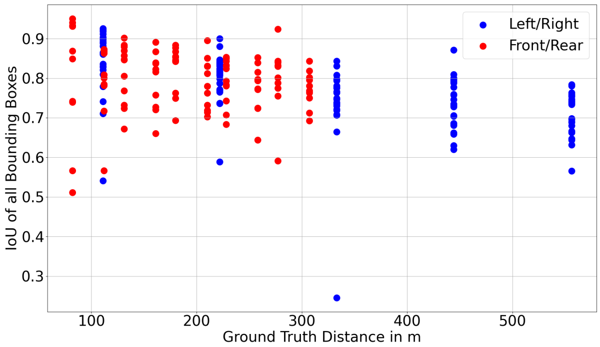

**FIG 8. IoU values in relation to the ground truth distance of the test samples. The colors distinguish between test sample with images from a front/rear view and from a side view.**

Figure 9 shows the IoU of each test sample in relation to the deviation between ground truth and the ensemble prediction average. It becomes visible that a lower IoU value does not necessarily mean a bad prediction. This is because the detection bounding boxes by the different networks may spread apart from each other leading to a low IoU while having a low deviation from ground truth after averaging. Anyway, since an uncertain prediction is not necessarily an inaccurate prediction, this does not say that IoU cannot be used as a criterion to assess uncertainty. A high IoU value between the bounding boxes indicates that the ensemble is jointly certain that the object is at that position. Therefore we propose looking at the IoU in order to estimate the uncertainty in an ensemble. The prerequisite for this is, of course, that the ensemble had shown both good precision and recall in the tests. However, how the IoU between all ensemble predictions can be integrated systematically and effectively into the uncertainty assessment should be investigated in future work.

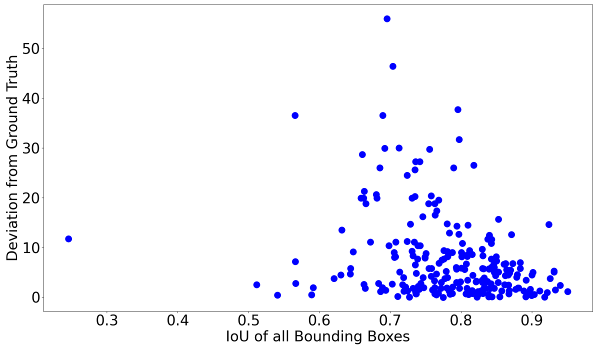

**FIG 9. IoU values in relation to the deviation from ground truth distance in m of the test samples.**

## 5. DISCUSSION

### 5.1. Performance

Although the rather simple ensembling approach of bagging was chosen, advantages of the ensemble could be shown. In terms of performance, the number of failed tests, i.e. undetected threats, was significantly reduced compared to the single network ( 4 compared to 21). Averaging the individual prediction results generally yielded more accurate results, especially test for test cases with longer distances. The networks trained on real-world images from the OpenImages data set performed overall well on the self-crafted FlightGear test set. However, the tests were designed to be simple (only one clearly distinct object in each pair of images) and did not include negative test cases, i.e. test cases where no object is present. Since ensembles also increase the probability of false positives of individual networks, such test cases should also be considered in future research. It would also be interesting to see how the ensemble would perform with video sequences instead of frames, as videos better represent real-world use of the system. Videos would also allow tracking information to be included in the uncertainty assessment. For our ensemble, the number of five networks was chosen rather arbitrarily. Finding the optimal number of networks in an ensemble is not straightforward and depends on the task and the availability of computing resources and high-quality training data. The number of networks must be high enough to ensure sufficient diversity, but as the number increases, the computational cost increases linearly, while beyond a certain point there is only marginal improvement or even performance degradation. Therefore, remaining accuracy deficiencies cannot be simply compensated by adding more ensemble members, but other strategies must be applied. Furthermore, the experiment used an ensemble of homogeneous detectors, as all five used the Yolov7 algorithm and architecture. For more diversity, it would be worth considering using different object detectors. For example, if time requirements permit, one-stage and two-stage detectors could be combined. Such an ensemble could benefit from the advantages of both detector types.

### 5.2. Redundancy

An important concept in safety is redundancy of components, so that in the event of a failure, a backup component can continue to perform the function. The safety provided by redundancy can be further enhanced by using dissimilar versions of that component in order to reduce the risk of common cause errors. In software development, this principle is called n-Version Programming. ED-12C/DO-178C [3] refers to this method as Multiple-Version Dissimilar Software. Neural network ensembles use multiple-version dissimilarity by default. By far the most common errors in neural networks are false predictions caused by the models weights. By combining different models that make errors on different subsets of the input space, errors of individual networks can often be compensated for [7]. This could also be observed during the studies on our distance-estimation system. In only four test cases could none of the five networks detect the potential threat. The average number of failed detections for all ensemble members was 20.6 on all test cases. The detector with the fewest failed detections missed to identify the aircraft in both images for 10 test cases. However, there still were many similarities among the members in our ensemble. By using different object detectors and independent data sets for training, the diversity in the ensemble could be increased. In general though, despite being an active research topic, there is no general consensus of how diversity should be measured and how diversity can most effectively be utilized in ensembles. Another argument in favor of using redundancy through ensembles is increased security against the most common attacks on neural networks. In poisoning attacks, malicious data is introduced into a training data set by an attacker. Manipulating all members of an ensemble simultaneously is difficult when distinct training data sets are used. In addition, the use of an ensemble can increase the resilience to the effects of a single manipulated member. Also, the independence of errors should lead to greater robustness against adversarial attacks [26], where an attacker causes small perturbations to the input data in order to cause false predictions. The individual members of the ensemble can also run on independent hardware, which further increases dependability. In this case a hardware fault of one instance will not result in a failure of the entire ensemble unless the aggregation unit is affected. It is also possible to identify faulty ensemble members by monitoring predictions and detecting frequent strong deviations from the average. Affected networks can then be given smaller weights in the decision-making process or be ignored completely.

While this paper has highlighted advantages of ensembles, there are also drawbacks of their usage. An obvious disadvantage of using ensembles is the higher effort for training and implementation and larger demand of computational power. To achieve the highest diversity, it would be necessary to create a separate independent training data set for each individual neural network such that no identical data samples appear in multiple sets. In practice, creating a single data set taking into account all quality features can already present a challenging and laborious task. Techniques, such as bootstrapping, can help to some extent but result in less independence between networks. For example, incorrectly annotated data in the training set can then affect multiple trained networks. Neural network verification and explainability which are essential for certification according to EASA's AI guideline [5] would also have to be applied to all individual models. This would significantly increase the complexity of system certification. Also the temporal aspect in ensembles must be considered. In a real-time application, care would have to be taken to ensure appropriate synchronization of all networks.

## 6. CONCLUSION

This paper is intended to motivate research on using ensembles for safety-critical perception tasks in aviation by highlighting desirable properties. It was demonstrated how an ensemble of five object detectors improved the performance of a conceptual distance-estimation system over a single detector. FlightGear was used to create a test set of different scenes for which the ensemble was used to determine the distance to an aircraft model. Compared to the single detector network, a smaller deviation from the ground truth value was observed on average for the ensemble. Furthermore, the ability of ensembles to estimate uncertainty in its predictions and the advantages of the inherent redundancy were highlighted. In our opinion the improved performance, the uncertainty estimation and the redundancy of ensembles are essential to achieve the best possible performance for AI-based systems in safety-critical computer vision tasks.

**Contact address:**

Jasper.Sprockhoff@dlr.de

### References

[1] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019. DOI: 10.1109/TNNLS.2018.2876865.

[2] Eric Jenn, Alexandre Albore, Franck Mamalet, Grégory Flandin, Christophe Gabreau, Hervé Delseny, Adrien Gauffriau, Hugues Bonnin, Lucian Alecu, Jérémy Pirard, et al. Identifying challenges to the certification of machine learning for safety critical systems. In *European congress on embedded real time systems (ERTS 2020)*, 2020.

[3] RTCA and EUROCAE. *Software Considerations in Airborne Systems end Equipment Certification*, 2011.

[4] EASA. Artificial intelligence roadmap 2.0: Human-centric approach to ai in aviation. Technical report, May 2023.

[5] EASA. Easa concept paper: First usable guidance for level 1 & 2 machine learning applications. Technical report, Feb. 2023.

[6] Valliappa Lakshmanan, Sara Robinson, and Michael Munn. *Machine learning design patterns*. O'Reilly Media, 2020.

[7] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990. DOI: 10.1109/34.58871.

[8] Ashnil Kumar, Jinman Kim, David Lyndon, Michael Fulham, and Dagan Feng. An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE journal of biomedical and health informatics*, 21(1):31–40, 2016. DOI: 10.1109/JBHI.2016.2635663.

[9] David West, Scott Dellana, and Jingxia Qian. Neural network ensemble strategies for financial decision applications. *Computers & operations research*, 32(10):2543–2559, 2005. DOI: 10.1016/j.cor.2004.03.017.

[10] Imran Maqsood, Muhammad Riaz Khan, and Ajith Abraham. An ensemble of neural networks for weather forecasting. *Neural Computing & Applications*, 13:112–122, 2004. DOI: 10.1007/s00521-004-0413-4.

[11] Luis A Ortega, Rafael Cabañas, and Andres Masegosa. Diversity and generalization in neural network ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 11720–11743. PMLR, 2022.

[12] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018. DOI: 10.1002/widm.1249.

[13] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object

detectors. *arXiv preprint arXiv:2207.02696*, 2022. DOI: 10.48550/arXiv.2207.02696.

[14] Alexander R Perry. The flightgear flight simulator. In *Proceedings of the USENIX Annual Technical Conference*, volume 686, 2004.

[15] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023. DOI: 10.1109/JPROC.2023.3238524.

[16] Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996. DOI: 10.1007/BF00058655.

[17] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[18] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. DOI: 10.1016/S0893-6080(05)80023-1.

[19] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021. DOI: 10.48550/arXiv.2107.03342.

[20] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021. DOI: 10.1007/s10994-021-05946-3.

[21] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[22] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[23] Jernej Mrovlje and Damir Vrancic. Distance measuring based on stereoscopic pictures. In *9th International PhD workshop on systems and control: young Generation Viewpoint*, volume 2, pages 1–6, 2008.

[24] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. DOI: 10.1007/s11263-020-01316-z.

[25] Ángela Casado-García and Jónathan Heras. Ensemble methods for object detection. In *ECAI 2020*, pages 2688–2695. IOS Press, 2020. DOI: 10.3233/FAIA200407.

[26] Thilo Strauss, Markus Hanselmann, Andrej Junginger, and Holger Ulmer. Ensemble methods as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1709.03423*, 2017. DOI: 10.48550/arXiv.1709.03423.