Analysis of YOLO neural network overfitting on precision of object detection

Danyil Klokta[1,†], Yuliya Averyanova[1,†] and Yevheniia Znakovska[1,*,†]

[1]Interregional Academy of Personnel Management, Frometivska str. 2, 03039, Kyiv, Ukraine

Abstract

Currently, detecting foreign object debris (FOD) on runways is a very important task for aviation to provide flight safety, regularity and economy. This, in turn, requires effective systems that can recognize these objects in real time. The promising instrument for this is the single-stage YOLO detectors. In this paper, the single-stage YOLOv8s detector is studied. Although they work quite effectively, they have several problems that need to be solved. One of them is the overfitting effect, which can remember noise. This effect results in the deterioration of the ability to recognize and classify objects. In this research, the impact of overfitting effect on the ability to detect and classify objects in the YOLO neural network was empirically analyzed. To call the overfitting effect, the conditions were changed manually to make worse conditions than the usual operating conditions for such a YOLOv8s model. The results showed that the overfitting effect was achieved. The large difference is in the losses (box loss) on the training set and on the validation set were observed. The losses on the training set constantly decreased and were in the range of 0.2, while the losses on the validation set stops in the range of 0.6 after 400 epochs and moved chaotically in this range. The model also has shown problems in classifying complex classes, such as lower accuracy than in basic classes. In the "BoltNutSettrack" class, the accuracy was 0.825. The results confirm that despite the large number of epochs and achieving maximum accuracy on the training set, the effect of overfitting can lead to false results on the validation set. The requirements to the safety in the aviation industry require to reduce the effect of overfitting. Therefore, the methods of regularization, augmentation, sufficient dataset size, and early stopping were considered.

Keywords

FOD detection, innovative technologies, industry innovation, resilient infrastructure, YOLOv8s, overfitting, neural networks, sustainable transport, aviation safety

1. Introduction

The problem of foreign object debris (FOD) on runways is still important for aviation safety [1]. FOD can cause damage to aircraft, lead to economic problems, and even accidents [2]. Traditional methods of detecting FOD on the runway require a lot of human resources and time. Moreover, as it is indicated in [3] the FOD detection systems that are most often used in the airports including those that are based on radar or visible light technology have limitations on the placement and can be susceptible to the wind and wind-related phenomena. There are some solutions for small airports that consider the placement of sensors onto the moving platform. In [3] it is indicated that there should be established technical standards concerning key performance indicators. In papers [4, 5, 6], the design features of control systems for moving platforms with equipment of different types are considered. The general recommendations on foreign object debris detection systems FODDS were also discussed in [7].

The latest advances in technologies including drones, computer vision, and artificial intelligence as well as fusion of data from different sensors becomes promising for task of FOD detection. In paper [8], the overview of the latest approaches to FOD detection is presented. It is indicated there that AI-based methods can be successfully used to identify FOD and non-FOD [9]. For example, paper [10] proposes and considers the method of FOD detection on the base of YOLOX architecture. The use of AI-based methods for object classification using aerial data also can be found in [11, 12, 13]. Neural networks for

WDA'26: International Workshop on Data Analytics, January 26, 2026, Kyiv, Ukraine

*Corresponding author.

†These authors contributed equally.

klokta.danyil@gmail.com (D. Klokta); ayua@kai.edu.ua (Y. Averyanova); evgeniya.znakovska@gmail.com (Y. Znakovska)

0009-0008-8089-7116 (D. Klokta); 0000-0002-9677-0805 (Y. Averyanova); 0000-0002-9064-6256 (Y. Znakovska)

object detection (such as yolo) show themselves quite promising, they have high speed and quite good precision for performing their tasks. It is necessary to note that the implementation of these innovative technologies and systems in the runway environment is connected with the range of difficulties. The first is the size of the foreign objects. They are very small compared to the runway. The next problems that should be taken into account when the tasks of FOD detection are the view of the runways and the conditions of their inspection. The runways are significantly mixcoloured surfaces that consist of different colors of the same gradient and have different covering textures. They are mostly unprotected from different lighting and weather conditions.

All these conditions create some restrictions to neural networks for foreign object detection, requiring them to be extremely accurate.

In aviation everything must be as accurate as possible. The neural network that can lead to such errors can cause flight delays, traffic management complications, and even accidents. Therefore, the reliability of a neural network is very important for the task of FOD detection. The model cannot just only remember different examples. It must summarize and generalize to identify objects correctly despite of the difficulties connected with the runway conditions.

Therefore, it is important to consider and resolve the problems connected with utilizing novel and innovative technologies, including AI-based technology, before the implementation in aviation industry. The neural networks, as many other novel technologies aimed to serve for resilient and sustainable infrastructures, have several disadvantages and problems that have to be solved. One of the disadvantages is connected with the phenomenon of the neural network overfitting. Overfitting can occur usually in cases of rather complex models and is revealed in the model remembering differing noises or some specific characteristics. The result is poor ability of the model to generalize the data correctly [14, 15]. As a consequence, the model produces much more errors on test compare to the training set. This results in a higher level of false detections when using the FOD neural model.

The purpose of this paper is the empirical analysis the overfitting of the You Only Live Once version 8 small (YOLOv8s) neural model on the dataset with special foreign objects on the runway. The analysis of how overfitting effect influence the prediction errors and how this effect can be critical for tasks of FOD detection on the runway is made by changing characteristics that are responsible for the overfitting effect, metrics, and the number of photos in the dataset.

2. Theoretical foundations and methodology

In this work, the You Only Look Once, version 8, small (YOLOv8s) model is used for object detection. The model selection is based considering the speed and precision for the purpose of our research [16]. This is a very important indicator, because FOD detection on runway should be done in a short time to satisfy the requirements of safety and taking into account the air traffic growth and airport capacity.

YOLOv8s is a single-stage detector (SSD). Compare to the two-stage detectors, they perform the detection task as a regression task [16]. Two-stage detectors also use a regression task. It is performed in two phases, that lead to greater precision, but need more time. In our case of a single-stage detector, the model simultaneously predicts classes and bounding box coordinates. Due to these peculiarities, the single-stage detectors are significantly faster than two-stage detectors, but they have slightly lower precision. This can become a problem when detection and classifying small objects that are located near each other.

Retinanet model was also considered as an option for object detection. The object detection using RetinaNet model and the method for distance calculation was proposed and analyzed in [17]. Retinanet model requires more time and computer resources. Our comparative analysis with RetinaNet [18] proves that YOLOv8s is the best choice for our object detection study.

Overfitting is appeared to be a deep problem in machine learning. It occurs with good adaptation of a model to the noise and very special characteristics of the training set. As a result, the model ability to generalize getting worse [19].

Overfitting is detected when the loss function still decreases on the training set, but increases on the validation set. We can define this difference with the formula (1). Let be the model parameters at epoch. Overfitting is detected, when [16]:

$$L_{train}(\Theta(t)) < L_{train}(\Theta(t-1)),\ L_{val}(\Theta(t)) > L_{val}(\Theta(t-1)) + \varepsilon, \quad (1)$$

where L_{train} is the training dataset precision loss, L_{val} is the validation dataset precision loss, is the allowed threshold.

This can be explained next way. The model continues to learn, and its precision improves on the training set, but the generalization error continues to increase, exceeding the allowed threshold.

This indicates that the model is starting to adapt to the training data, but at the same time its performance and ability to analyze new real data are getting worse. This can lead to object gap errors [19, 20].

To prevent overfitting, the ability to early-stop was added starting with YOLOv4. Tests showed that the number of epochs decreased, while precision did not become the worse [21].

The several studies in this field confirms the relevance of the result of overfitting for YOLO. The ability of the model to remember background noise is connected with a complex background [21]. This is typical for the common runways, especially when the task is to identify the dark objects on a dark background. The data augmentation methods are used to avoid this. Data augmentation can be defined as a set of methods to change the characteristics of an object by transformation the photographs and objects on it.

The problem of overfitting also arises when the FOD object is small. Detection of small objects is probably the weakest point of YOLO. It requires careful control of the training parameters and the separation of the dataset [22]. This is necessary in order to generalize the ability to analyze and avoid noise holding in the training data. Therefore, overfitting analysis is an important part of using the YOLO model in FOD object detection, as well as for all neural networks in general.

3. Materials and methods

We used a specialized dataset taken from the Roboflow platform to study and evaluate the modeling of foreign objects (FOD) [23]. The dataset comprises of runways images and foreign objects on them. The total number of images in the dataset is 3560, which were divided into three subsets: training (3214 images), validation (172 images), and testing (approximately 174 images). Each image was signed and annotated for 11 specific FOD classes. The images mostly have a resolution optimized for training the models, which allows for efficient detection of even small objects. The dataset was reduced to 300 images to achieve the overfitting effect.

The images have the typical complexity of the runway background, including asphalt surfaces and markings. The dataset covers different training conditions, including daylight or low light. This impaired the generalizability of the model. There is also different variation in the size of FOD objects. This is important for evaluating the ability of models to detect small objects or objects of different sizes. Annotations for FOD objects are provided in the corresponding folders. This dataset has realistic photos that is important for FOD research. All images were scaled to a resolution of 416x416 pixels for training [1].

The YOLOv8s model (YOLOv8 Small) was selected and adapted for training. The model was firstly trained on a large dataset called Common Objects in Context (COCO) which has 80 classes of objects [24]. This allowed us to use already trained features and parameters for our images as well to save time and continue with the training process.

The realization and training of the YOLOv8s model were performed using the PyTorch framework and the Ultralytics YOLOv8 library. The training process was performed in the Google Collaboratory cloud environment. For faster training we used a 16 GB NVIDIA Tesla T4 Graphics Processing Unit (GPU). The stochastic gradient descent (SGD) algorithm was used to optimize the network weights.

4. Conditions for provoking overfitting

In order to empirically demonstrate the effect of overfitting, the YOLOv8s model was purposely trained under conditions that should cause overfitting, such as:

1. Increased number of epochs: The number of epochs was increased to 1000 epochs, accordingly, which usually exceeds the optimal number required for training. This allows us to fix the beginning of the model mistakes on the validation set. After this, it is possible to observe the learning process with a higher number of mistakes on the validation set than on the training set.
2. Disabled augmentation: Any data augmentation methods were disabled. Augmentation usually allows the model to increase its generalization ability; its disabling gives us faster memorization of specific features of the training set.
3. Disabled the built-in Ultralytics early-stopping function: The early-stopping function was disabled in our training process. The YOLOv8s model can stop training if the model stops improving for a certain number of epochs (100 epochs).
4. Dataset reduction: To induce the overfitting effect, the dataset was reduced to 300 images and their labels were reduced accordingly. This reduction of the number of images leads to an almost exact one-hundred percent achievement of the overfitting effect, which allows us to conduct further analysis.
5. Reduction of the penalty for large weights: The penalty for large weights was reduced to 0, which also has an influence on the overfitting effect, since under normal conditions, this parameter allows the model to optimize its weights a little better. In our case, we have no penalty for large weights, which allows the model to create a very complex structure. This, in turn can lead to good weights for 300 images from the dataset, but at the same time to quite a lot of errors on the validation set. This is because the model remembered the noise and features of the training data.

The next metrics were used for learning process analysis and overfitting detection: Bounding Box Loss, object classification loss, precision, average precision, mean average precision, the normalized error matrix and the precision-Recall Curve.

5. Empirical demonstration and results

For a correct estimation of the learning process of the YOLOv8s model and empirical demonstration of overfitting detection, the main components were analyzed on the FOD training and validation datasets. These include: three main loss metrics, precision, average precision, mean average precision, the normalized error matrix and the precision-Recall Curve.

The YOLOv8 loss function commonly has three main loss metrics.

1. Box Loss evaluates the precision of detecting the space location and size of the objects.
2. Distributed Focal Loss is a special loss function used for accurate frame location. It is responsible for centering the frame in the middle of the object.
3. Class Loss is the metric that is responsible for classifying objects (e.g., a nut or a screw). It compares the data with the corresponding labels provided at input.

To evaluate the learning process L_{total} (2), we will use the loss function used in YOLOv8, which is calculated as a weighted sum of these three metrics [16]:

$$L_{total} = \lambda_{box} L_{box} + \lambda_{cls} L_{cls} + \lambda_{dfl} L_{dfl}, \tag{2}$$

where L_{box} is the Complete Intersection over Union (CIoU). CIoU aims to reduce the error between the predicted frame and the real one; L_{cls} is the classification loss for correct object classification; L_{dfl} means (Distributed Focal Loss) and defines the frame borders.

Theoretically, minimization of L_{total} leads to accurate detection. However, as we demonstrate below, minimizing L_{train} does not give us guaranteed performance on new data.

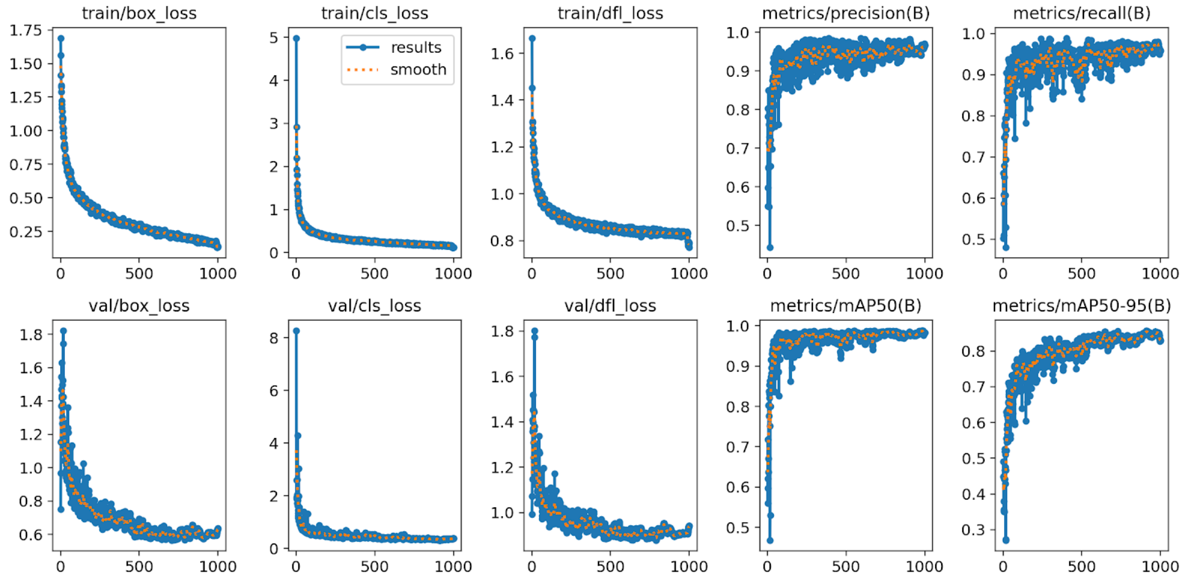

Figure 1: The results of overfitting YOLOv8s.

As the results of our experiments show the largest error in determining the object's bounding box, it is important to determine the Box Loss function (L_{box}) [16]. YOLOv8 uses a CIoU to reduce the geometric error (2):

$$L_{box} = 1 - \text{IoU} + \frac{\rho^2(b, b^g)}{c^2} + \alpha v, \tag{3}$$

where IoU is the Intersection over Union ratio; b and b^g indicate the center points of the predicted box and the ground truth box correspondingly; $\rho(\cdot)$ is the Euclidean distance between these center points; c is the diagonal length of the smallest box that covers both boxes; and α is a penalty for incorrect aspect ratio recognition.

The difference of this metric on the validation set tells us that the model cannot determine the precise location of our objects.

In Figure 1 the overall estimation of our training is shown. The graphs show the relationship between the epochs and loss precision. The first graph demonstrates the box losses on the training part of our dataset. It is possible to find in Figure 1, that Box Loss constantly decreases in case of the training data, but in case of the validation data this indicator moves to a value of 0.6 then stops and continues to move very chaotically. Then, it demonstrates the very jumpy movements during all training process. Moreover, in the end part of the validation box loss graph (first one in the second row) on the validation set we have the losses that begin to grow, while on the training set (first graph in the first row) they still continue to decrease. As a result, we have a large gap between 0.2 (training part) and 0.6 (test part). The model predicts good results on familiar data, but on new data it shows frames 3 times worse.

Let us to consider Distributed Focal Loss (third graph in the first and the second row). We can observe the similar result. On the training data, the Distributed Focal Loss constantly decreases. But the validation data after about 300 epochs demonstrates the chaotic movement with significant vertical fluctuations (from 0,5 to 1,2) at the beginning. The difference between the validation and training data can be seen as the smoothed value of the losses.

All other metrics precisions, precision and mean Average Precision (mAP) 50-95 show the similar chaotic movement, but still identify the object correctly in most cases. The results show that the YOLOv8s model is quite accurate in ideal situation identifying objects even with considered parameters.

The training performance of the YOLO model is also evaluated using Precision and Recall methods. The formulas themselves look like this [16]:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}, \tag{4}$$

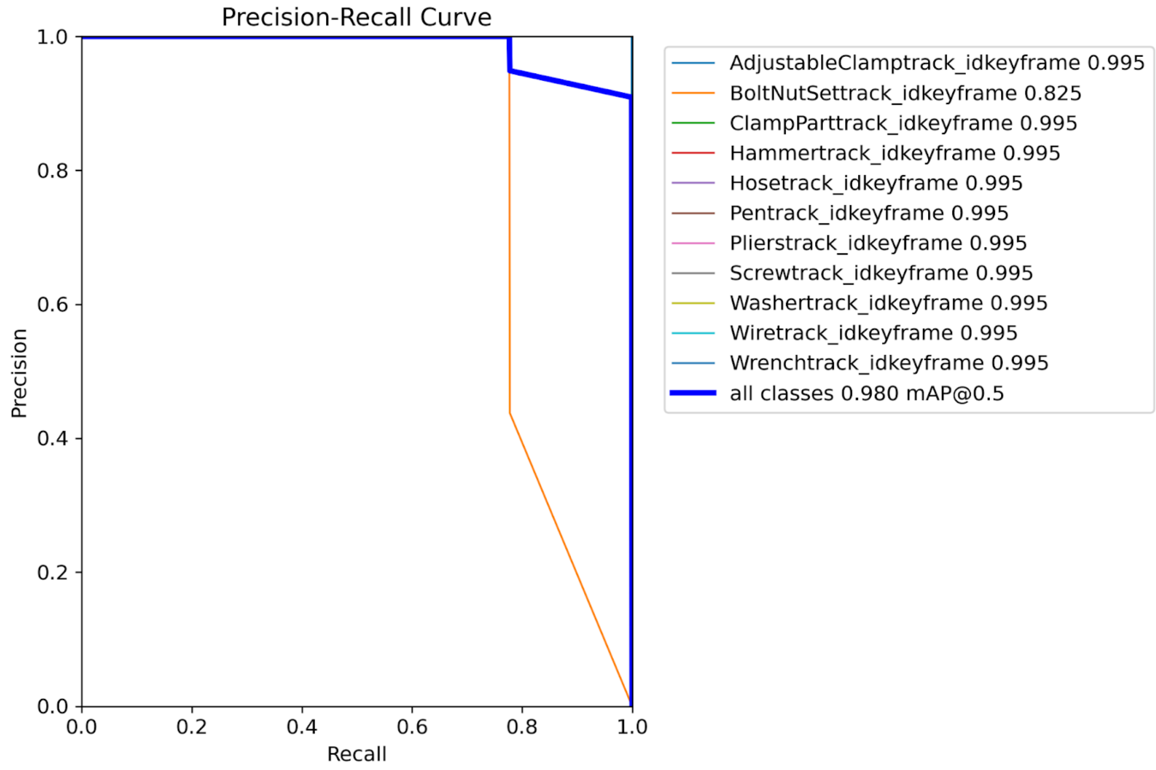

Figure 2: The precision-recall curve.

where TP means True Positives, FP means False Positives, and FN means False Negatives.

The Precision-Recall curve is given in Figure 2. It shows the relationship between these indicators through calculated values. The overall performance is 0.980. This value demonstrates rather good result for our considered situation. But there is one class that shows weakness, which is BoltNutSettrack_idkeyframe. It is marked by the orange line, and has a much lower average Precision (AP) (0.825) than the other classes (almost all 0.995).

The curve for this class drops off sharply when Recall exceeds about 0.8. This means that the model starts making a lot of false positive forecasts (Fps), trying to find the last 20% of true objects in this class. This means that the model is not identifying this class well enough.

Let us analyze the normalized error matrix. It is presented in Figure 3. It shows us quite good precision considering our parameters. The majority of values equals 1. This indicates accurate classification of objects. Despite this, we have problems in some classes. These are clearly seen in the Wiretrack_idkeyframe and Hummertrack _idkeyframe classes, which give us a precision error of 37%. We also have a precision error of 11-12% for BoltNutSettrack_idkeyframe, ClampParttrack_idkeyframe and Pentrack_idkeyframe.

We can conclude from the analysis of the normalized error matrix that in the most cases the overall precision is high and equals 100%. But for some classes it has a performance of 70-80%. Commonly, this is not bad precision. But for the task of FOD detection on a runway, this is not a sufficient indicator as the correctness of FOD detection directly influence the aviation safety. Especially, when we can achieve the better precision by disabling the parameters which can lead to overfitting.

Figure 4 shows the results.csv file and demonstrates the proof of overfitting. In Figure 4 the Box Loss metric on the training data is constantly decreasing. The relatively low values in Figure 4, such as: 0.255–0.265 mean rather good results. However, the Box Loss metric on the validation data after 400 epochs begins to growing up and reaches the values of 0.58 – 0.62. It is approximately 2,5 times higher than on the training set.

The box loss means that the drone can find some foreign object debris, but it can be false and

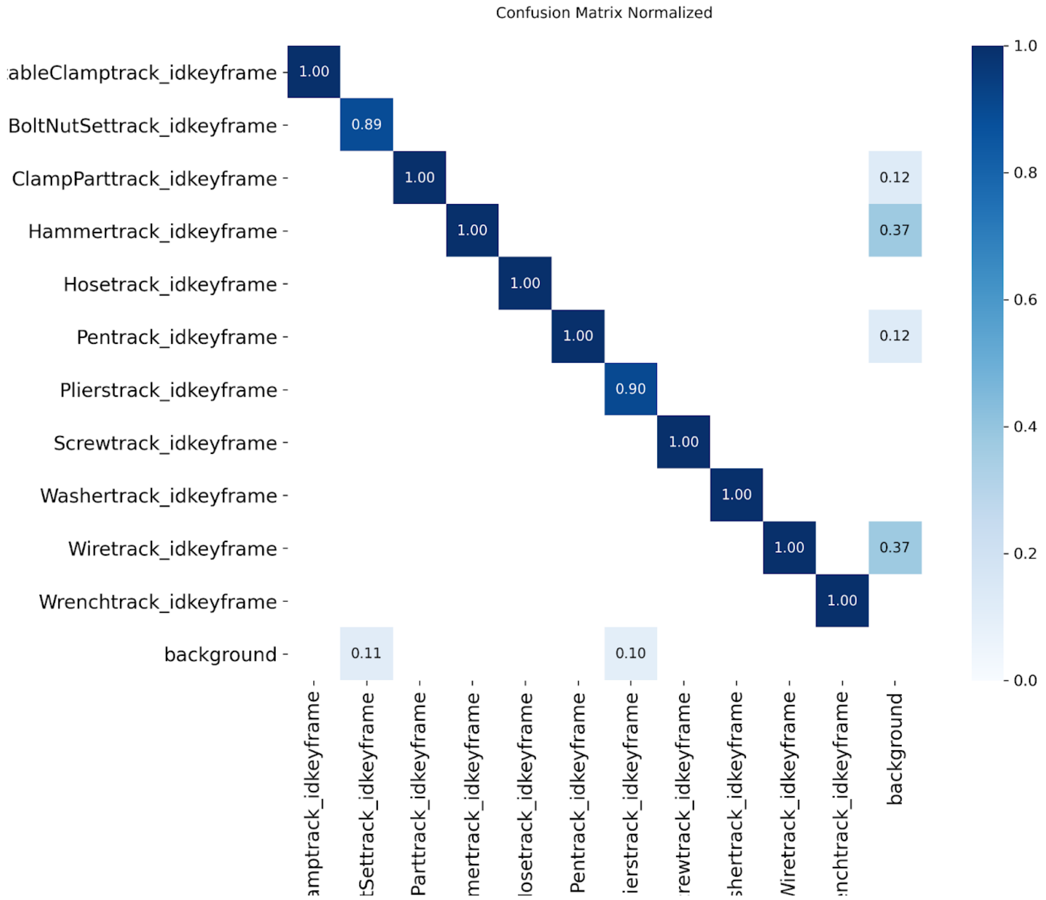

Figure 3: The normalized error matrix.

499	2976,1	0,30666	0,28091	0,86738	0,91098	0,8844	0,95078	0,7795	0,69282
500	2983,2	0,28959	0,25766	0,85498	0,95784	0,84036	0,95269	0,77506	0,69261
501	2988,7	0,28569	0,26186	0,85638	0,93964	0,87244	0,94579	0,77344	0,68226
502	2994,7	0,29264	0,27063	0,85561	0,9362	0,86916	0,94584	0,78322	0,66904
503	3001	0,27779	0,24999	0,85436	0,89982	0,89394	0,94584	0,77618	0,66566
504	3006,5	0,27032	0,25032	0,84512	0,94509	0,89165	0,96515	0,79631	0,64059
505	3013,4	0,28463	0,26199	0,84486	0,97061	0,89698	0,9708	0,80558	0,6246
506	3018,8	0,27021	0,24444	0,8509	0,93781	0,8937	0,96295	0,80709	0,6241
507	3025,4	0,26746	0,24863	0,84604	0,88161	0,9223	0,95121	0,80107	0,63194
508	3031,2	0,30583	0,26401	0,84627	0,91254	0,91436	0,95584	0,80201	0,61925
509	3036,8	0,27999	0,25321	0,84946	0,95997	0,88943	0,95976	0,79504	0,62396
510	3043,3	0,28088	0,25028	0,84657	0,95834	0,95806	0,98338	0,82015	0,6017
511	3048,9	0,27231	0,25408	0,85391	0,95151	0,96106	0,9842	0,83403	0,60062
512	3055,9	0,27886	0,24715	0,84162	0,97173	0,95361	0,98623	0,84139	0,58997
513	3061,5	0,29188	0,26775	0,85093	0,95931	0,96126	0,98199	0,83065	0,60025
514	3067,5	0,27087	0,24466	0,84543	0,96278	0,95255	0,98253	0,83402	0,60946
515	3073,7	0,27836	0,24575	0,85104	0,94067	0,96151	0,98034	0,82666	0,61593
516	3079,2	0,26555	0,2489	0,85228	0,97543	0,95148	0,98179	0,82526	0,61313
517	3086	0,26517	0,2475	0,84307	0,95809	0,96289	0,98346	0,83175	0,60653
518	3091,6	0,25572	0,24005	0,83837	0,94117	0,94808	0,9794	0,81882	0,63194
519	3098,2	0,26541	0,24391	0,8434	0,94776	0,90361	0,97201	0,81216	0,65198
520	3104,3	0,26581	0,24214	0,84295	0,95366	0,88647	0,95056	0,78416	0,68615
521	3109,8	0,27853	0,25385	0,85628	0,96199	0,90128	0,96406	0,78236	0,66797

Figure 4: The part from the results .csv file.

inaccurate in determination of their location. It can result in the necessity to repeat the runway check by the ground crews or to make additional check with the drone in the same area. Obviously it is the waste of resources and time.

Figure 5 shows the cls loss. The results of Figure 5 are seemed as not effective again. The validation of cls loss is also reduces training cls loss, then it is stopped about the values of 0.35 – 0.39. While the training cls loss has the trend to decrease, the validation cls loss stops decreasing after 700 epochs. This

A	B	C	D	E	F	G	H	I	J	K
722	4333	0,21677	0,20724	0,83956	0,94188	0,96878	0,98385	0,83879	0,5995	0,37611
723	4340	0,22754	0,21621	0,8475	0,93905	0,95565	0,9823	0,84468	0,58736	0,36802
724	4345,5	0,2138	0,19704	0,83361	0,92652	0,96535	0,98209	0,84465	0,56812	0,3535
725	4352	0,2254	0,21134	0,83123	0,94102	0,96503	0,98492	0,85229	0,56998	0,36168
726	4358,1	0,21214	0,20787	0,83122	0,93261	0,96209	0,98206	0,85286	0,58741	0,37116
727	4363,8	0,21591	0,2048	0,83535	0,92356	0,97464	0,98272	0,85521	0,5918	0,37124
728	4370,8	0,20443	0,18746	0,83489	0,92376	0,96755	0,9818	0,85161	0,56981	0,34878
729	4376,2	0,22666	0,20747	0,84215	0,93219	0,96264	0,98128	0,84708	0,57071	0,34992
730	4383,2	0,20861	0,20323	0,83268	0,92052	0,97482	0,98322	0,83998	0,58078	0,34931
731	4389,1	0,21529	0,21163	0,84948	0,9527	0,96773	0,98396	0,84258	0,5766	0,3486
732	4395,7	0,20374	0,19454	0,83802	0,95233	0,96973	0,98417	0,84598	0,58122	0,35907
733	4401,9	0,2104	0,19141	0,82955	0,95676	0,97776	0,98519	0,84685	0,5734	0,35024
734	4407,5	0,2227	0,21436	0,83146	0,95194	0,97713	0,98446	0,84646	0,57196	0,35544
735	4414,7	0,21925	0,21367	0,83645	0,94376	0,9742	0,98389	0,84588	0,57757	0,35956
736	4420,3	0,21315	0,19935	0,83685	0,9455	0,96768	0,98303	0,83999	0,57727	0,35793
737	4426,8	0,22561	0,20881	0,829	0,94831	0,97018	0,98475	0,83935	0,5757	0,36684
738	4432,9	0,22806	0,21541	0,84185	0,94863	0,95753	0,98395	0,8397	0,5746	0,37864
739	4438,7	0,22758	0,2097	0,8402	0,96704	0,93687	0,98031	0,83636	0,58537	0,384

Figure 5: The part from the results .csv file with cls loss.

can mean that some objects are not classified correctly. The same can be seen in the Figure 2 and Figure 3. Therefore, we can conclude that some objects, especially small ones, cause some difficulties for the neural model.

The discussed indicator can be considered as direct empirical proof of overfitting. It shows that the model adaption to noise and the specific peculiarities of the training set. But the generalization on the validation data continues to deteriorate. The research on loss precision demonstrates that the model overfitting means worse predictions of the actual position and classification of FOD objects. In case of use in the aviation industry, this is unacceptable as means decrease of safety.

6. Conclusions

In this research, we studied the effect of overfitting on the YOLOv8s neural model applied to foreign object detection on the runway. The overfitting was achieved empirically. The obtained results show the relationship between overfitting and quality classification. The analysis of the real data was made. To study the effect of overfitting, the parameters of the YOLOv8s model were changed. These include: a huge number of epochs (up to 1000 epochs), turning off augmentation and early stopping (Early Stopping), and absence of regularization (reducing the weight penalty to 0). The dataset was reduced to 300 images. The main results confirm the overfitting hypothesis. The main metrics loss has shown a steady decrease of error on the training set. It also shows the chaotic movement and cease of the error at values (0.58-0.62) on the validation part of the dataset. This can be considered as the direct proof of the loss of model generalization skill and the fact that model remembers the noise and specific features of the training dataset.

The other important metrics were studied. They include: the precision-Recall Curve, the normalized error matrix, precision, average precision, mean average precision. Their study and analysis demonstrate that the overfitting effect has been achieved. The obtained result indicates that it is important to control the parameters of the neural model and use methods that allow to avoid overfitting (for example, early stopping and augmentation). This is critically important for the safety of aviation when FOD detection.

Also, we emphasize that training after the optimal number of epochs (about 150-200) does not give any better results. At the same time, longer training may require larger machine resources. This confirms the fact that determining the optimal number of epochs is also very important for optimizing the time and energy spent on training. Furthermore, overfitting influences the classification of complex classes such as "BoltNutSettrack" in rather poor manner. As it is shown in the Precision-Recall curve analysis, we have an average precision of 0.825. This indicates that even the overtraining effect, such as memorizing noise and very fine details of objects, does not allow to recognize correctly a sufficient number of details of the complex class.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] ICAO, Icao doc 10004: Global aviation safety plan 2023-2025, 2023. URL: https://www2023.icao.int/safety/GASP/Documents/10004_en.pdf.

[2] J. Shan, L. Miccinesi, A. Beni, L. Pagnini, A. Cioncolini, M. Pieraccini, A review of foreign object debris detection on airport runways: Sensors and algorithms, Remote Sensing 17 (2025). URL: https://www.mdpi.com/2072-4292/17/2/225. doi:10.3390/rs17020225.

[3] ICAO, The application of fod detection equipment on airport pavement, 2025. URL: https://www.icao.int/sites/default/files/Meetings/a42/Documents/WP/wp_598_en.pdf.

[4] O. Sushchenko, Y. Bezkorovainyi, O. Solomentsev, M. Zaliskyi, O. Holubnychyi, I. Ostroumov, Y. Averyanova, V. Ivannikova, B. Kuznetsov, I. Bovdui, T. Nikitina, R. Voliansky, K. Cherednichenko, O. Sokolova, Algorithm of determining errors of gimballed inertial navigation system, in: O. Gervasi, B. Murgante, C. Garau, D. Taniar, A. M. A. C. Rocha, M. N. Faginas Lago (Eds.), Computational Science and Its Applications – ICCSA 2024 Workshops, Springer Nature Switzerland, Cham, 2024, pp. 206–218.

[5] M. Zaliskyi, O. Solomentsev, O. Holubnychyi, I. Ostroumov, O. Sushchenko, Y. Averyanova, Y. Bezkorovainyi, K. Cherednichenko, O. Sokolova, V. Ivannikova, R. Voliansky, B. Kuznetsov, I. Bovdui, N. Tatyana, Methodology for substantiating the infrastructure of aviation radio equipment repair centers, in: I. Ostroumov, K. Slimani, M. Zaliskyi, T. L. (Eds.), CEUR Workshop Proceedings, volume 3732, CEUR-WS, 2024, pp. 136–148. URL: https://ceur-ws.org/Vol-3732/paper11.pdf.

[6] O. Solomentsev, M. Zaliskyi, O. Holubnychyi, I. Ostroumov, O. Sushchenko, Y. Bezkorovainyi, Y. Averyanova, V. Ivannikova, B. Kuznetsov, I. Bovdui, T. Nikitina, R. Voliansky, K. Cherednichenko, O. Sokolova, Efficiency analysis of current repair procedures for aviation radio equipment, in: I. Ostroumov, M. Zaliskyi (Eds.), Proceedings of the 2nd International Workshop on Advances in Civil Aviation Systems Development, Springer Nature Switzerland, Cham, 2024, pp. 281–295.

[7] ICAO, Enhancing aviation safety and efficiency: Recommendation for foreign object debris detection systems (fodds), 2024. URL: https://www.icao.int/sites/default/files/APAC/Meetings/2024/2024%20AP-ADO-TF-5/3-Working%20Papers/WP14-AI-8-AAPA-FSF-FODDS-Paper-final.pdf.

[8] A. G V, P. R, The modern approaches for identifying foreign object debris (fod) in aviation, in: 2024 International Conference on Integrated Circuits and Communication Systems (ICICACS), 2024, pp. 1–5. doi:10.1109/ICICACS60521.2024.10498909.

[9] D. Klokta, Y. Znakovska, Y. Averyanova, Unmanned technologies and drones in runway service, in: O. Lytvynov, V. Pavlikov, D. Krytskyi (Eds.), Integrated Computer Technologies in Mechanical Engineering - 2024, Springer Nature Switzerland, Cham, 2025, pp. 230–241.

[10] J. Taupik, T. Alamsyah, A. Wulandari, E. U. Armin, A. Hikmaturokhman, Airport runway foreign object debris (fod) detection based on yolox architecture, in: 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE), 2023, pp. 40–43. doi:10.1109/ICCoSITE57641.2023.10127676.

[11] P. Prystavka, K. Dukhnovska, O. Kovtun, O. Leshchenko, O. Cholyshkina, V. Semenov, Recognition of aerial photography objects based on data sets with different aggregation of classes, Eastern-European Journal of Enterprise Technologies 1 (2023) 6–13. URL: https://journals.uran.ua/eejet/article/view/272951. doi:10.15587/1729-4061.2023.272951.

[12] V. Zivakin, O. Kozachuk, P. Prystavka, O. Cholyshkina, Training set AERIAL SURVEY for data recognition systems from aerial surveillance cameras, in: A. Anisimov, V. Snytyuk, A. Chris, A. Pester, F. Mallet, H. Tanaka, I. Krak, K. Henke, O. Chertov, O. Marchenko, S. Bozóki, V. V. Tsyganok, V. Vovk (Eds.), Selected Papers of the IX International Scientific Conference "Information Technology and Implementation" (IT&I-2022). Conference Proceedings Kyiv, Ukraine, November

30 - December 02, 2022, volume 3347 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 246–255. URL: https://ceur-ws.org/Vol-3347/Paper_21.pdf.

[13] P. Prystavka, S. Dolgikh, O. Cholyshkina, O. Kozachuk, Latent representations of terrain in aerial image classification, in: V. Ermolayev, D. Esteban, H. C. Mayr, M. Nikitchenko, S. Bogomolov, G. Zholtkevych, V. Yakovyna, A. Spivakovsky (Eds.), Proceedings of the 17th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I: Main Conference, PhD Symposium, and Posters, Kherson, Ukraine, September 28 - October 2, 2021, volume 3013 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 86–95. URL: https://ceur-ws.org/Vol-3013/20210086.pdf.

[14] A. Haj Hassine, K. Hamed, A Comparative Analysis of YOLOv11 Models: Impact of Object Size on Detection Performance, Degree project in computer science and engineering, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, 2025. URL: https://www.diva-portal.org/smash/get/diva2:1985752/FULLTEXT01.pdf.

[15] GeeksforGeeks, Machine learning: Underfitting and overfitting in machine learning, 2025. URL: https://www.geeksforgeeks.org/machine-learning/underfitting-and-overfitting-in-machine-learning/.

[16] Ultralytics, Yolov8: State-of-the-art computer vision model, 2025. URL: https://yolov8.com/.

[17] M. Al-Hasanat, M. Alsafasfeh, A. Alhasanat, S. Althunibat, Retinanet-based approach for object detection and distance estimation in an image, International Journal on Communications Antenna and Propagation (IRECAP) 11 (2021) 19. doi:10.15866/irecap.v11i1.19341.

[18] Y. Li, A. Dua, F. Ren, Light-weight retinanet for object detection on edge devices, in: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), 2020, pp. 1–6. doi:10.1109/WF-IoT48130.2020.9221150.

[19] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. doi:10.1109/CVPR.2016.91.

[20] T. Nikitina, B. Kuznetsov, O. Sushchenko, I. Ostroumov, N. Kuzmenko, M. Zaliskyi, O. Solomentsev, N. Ruzhentsev, O. Havrylenko, A. Popov, V. Volosyuk, O. Shmatko, S. Zhyla, V. Pavlikov, E. Tserne, K. Dergachov, Method for design of magnetic field active silencing system based on robust meta model, in: S. Shukla, H. Sayama, J. V. Kureethara, D. K. Mishra (Eds.), Data Science and Security, Springer Nature Singapore, Singapore, 2024, pp. 103–111.

[21] A. R. Muhammad, H. P. Utomo, P. Hidayatullah, N. Syakrani, Early stopping effectiveness for yolov4, Journal of Information Systems Engineering and Business Intelligence 8 (2022) 11–20. URL: https://e-journal.unair.ac.id/JISEBI/article/view/29988. doi:10.20473/jisebi.8.1.11-20.

[22] T. Schneidereit, S. Gohrenz, M. Breuß, Object detection characteristics in a learning factory environment using yolov8, 2025. URL: https://arxiv.org/abs/2503.10356. arXiv:2503.10356.

[23] ZeeAiTech, Fod-dataset dataset, 2022. URL: https://universe.roboflow.com/zeeaitech/fod-dataset.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 740–755.