sciendo

Transport and Telecommunication, 2025, volume 26, no. 2, 103-117
Transport and Telecommunication Institute, Lauvas 2, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2025-0009

AIRCRAFT PILOT ASSISTANCE SYSTEM: DETECTION OF OTHER AIRCRAFT USING ARTIFICIAL NEURAL NETWORKS

Paweł Tomiło

Lublin University of Technology
Faculty of Management, Lublin, Poland
p.tomilo@pollub.pl

The publication presents a pilot assistance system that is based on video camera frames and uses the developed 3AFPBB-YOLO artificial neural network model. The model is able to determine the relative location of other aircraft in the vicinity of an aircraft equipped with this system. Current solutions, the model development process, training, validation, and testing are presented. In the development process, different types of solutions as well as sizes of detection models were tested. In addition, the model inference time with the use of a single-chip mini-computer was examined. The developed model has better values of metrics compared to other solutions. A number of tests were conducted both using a ground station and actual use of the system in flight. The tests conducted using the ground station and the auxiliary hyper-inference algorithm showed that the effective distance from which the model is able to detect an aircraft is 705.09 m. At a fixed distance, the model's effectiveness is 90.14%. In the flight test using the auxiliary tracking algorithm, the model's effectiveness was 100%. The developed model is capable of real-time inference.

Keywords: Artificial neural network, aviation safety, assistance system

1. Introduction

Aircraft collisions may result from the intensifying air traffic brought on by aviation's progress. It gets harder to control air traffic and keep safe distances between aircraft as the number of aircraft using the airspace grow. Increased traffic raises the possibility of crashes during takeoff and landing operations, both on the ground and in the air. Maintaining flight safety in general aviation requires a thorough situation awareness of the surrounding environment. Therefore, it is becoming necessary to implement advanced air traffic management systems that help monitor aircraft positions and alert pilots to potential dangers. Such solutions include:

- TCAS - Traffic Collision Avoidance System (Tang, 2017);
- ADS-B - Automatic Dependent Surveillance-Broadcast (Yang *et al*., 2023);
- On-Board RADAR (Maas *et al*., 2020);
- Systems using aviation communication network (Džunda *et al*., 2022).

TCAS is a popular and effective method for reducing the likelihood of aircraft collisions. This tools are actually a family of airborne equipment that operate independent of the ground-based air traffic control (ATC) system. TCAS version II provides traffic advisories as well as resolution advisories, which can include recommendations for escape maneuvers. The main components of the device are the computer unit and the transponder. The computer unit is responsible for processing sensor data and controlling the collision avoidance logic. Transponder is responsible for communication - receiving and transmitting signals. The on-board device does not make decisions on its own but only displays a suggestion to pilots (Federal Aviation Administration, 2011; Tang, 2017). Another method that makes flight surveillance possible is ADS-B. Its foundation is the usage of a device that tracks its location by satellite navigation or other sensors, then broadcasts that position and other relevant data on a regular basis to allow for tracking. The information can be received by ATCs or satellite-based receivers (Yang *et al.*, 2023).

Even though the majority of light aircraft have Global Position System (GPS) receivers, their pilots are unable to determine the location of other flying objects that are in close proximity to them. Light airplanes usually do not have the anti-collision system installed. Only a small percentage of them use the more recent ADS-B technology to track the positions of other aircraft in their vicinity (Džunda *et al*., 2022).

One solution that can assist pilots is Frequency Modulated Continuous Wave (FMCW) radar. This device is responsible for creating a continuous signal whose frequency varies relative to a central value. Radar sends out a signal, which is reflected off the surfaces of objects in its vicinity and is then received by antennas. Based on the differences that occur between the transmitted and received signals, delay calculations can be made and the distance can be determined from them. Using this method, it is possible to determine the distance to the target and the radial velocity of the target. FMCW radars are not expensive, due to the reading of lower frequencies, energy consumption is much lower than that of pulse radar. By using FMCW radar, the object the vicinity of an aircraft can be taken into account to predict other aircraft, where any conflicts occurring between the speed of one's own aircraft and the routes of other aircraft can be noted at an earlier stage of forecasting (Fursich *et al*., 2016; Maas *et al*., 2020).

(Džunda *et al.*, 2022) presented a concept for an Anti-Collision System for Small Civil Aircraft that operates on the premise that all flying objects are part of the aviation communication network and send positional information. Another premise is that that the communication network must be synchronized, as well as each user has a specific time in which information transmission must be made. The system is based on determining the unknown point P_{un} based on knowledge of the location of a minimum of four other points and their distance to P_{un}.

Artificial neural network (ANN) models are used in a wide range of fields, from signal processing (Kulisz *et al.*, 2024) to object detection (Kateb *et al.*, 2021) through generative models (Rombach *et al*., 2021). Therefore, in order to increase air traffic safety, it was decided to develop a solution that would provide pilots with information regarding the location of other aircraft that are in their vicinity. The discussed solution is aimed at use in light aircraft for training and tourism purposes. The solution is based on the use of a video camera placed on the aircraft. The frames from the camera are analyzed by an artificial neural network model, which is based on the YOLO architecture. A light airplane, at a great distance from the camera is represented by a few pixels for this reason in the architecture it is necessary to use solutions that will allow the detection of such objects, while keeping in mind that model must inference in real-time. Therefore, the model architecture uses P2-level features along with Neck in the form of AFPN. The developed model was named 3AFPBB-YOLO. The described solution can help locate other aircraft without the need for each one to have the right device.

1.1. Overview of You Look Only Once (YOLO) architecture

The Yolo family models share a similar architecture, with the three primary parts — the Head, Neck, and Backbone — distinguishable in each instance. Depending on the YOLO model version, each of these components is in charge of a distinct kind of activity and has a variable number and types of layers that contribute to it. The foundation of Backbone is the idea of receptive cells in each case. The yolo models typically use feature levels from 1 to 5 $\{P_1, P_2, P_3, P_4, P_5\}$, with the standard version using data from levels 3-5 $\{P_3, P_4, P_5\}$. The designation P_n specifies the stride s (the element number used to move the filter throughout the input tensor) for the output of a given block, such that $s = 2 \cdot n_p$, where n_p is the tuple (n, n) for a given P_n, and the $\cdot$ operator denotes element-wise multiplication. The features gleaned from the various levels of the backbone model are combined by the Neck. The YOLOv8 architecture uses Neck in the form of a Feature Pyramid Network (FPN). The last component is Head, which is responsible for the final predictions in the form of bounding boxes and classes including confidence scores (Tan *et al*., 2020; Terven and Cordova-Esparza, 2023). A schematic of the simplified architecture of the yolov8 model is shown in Figure 1.

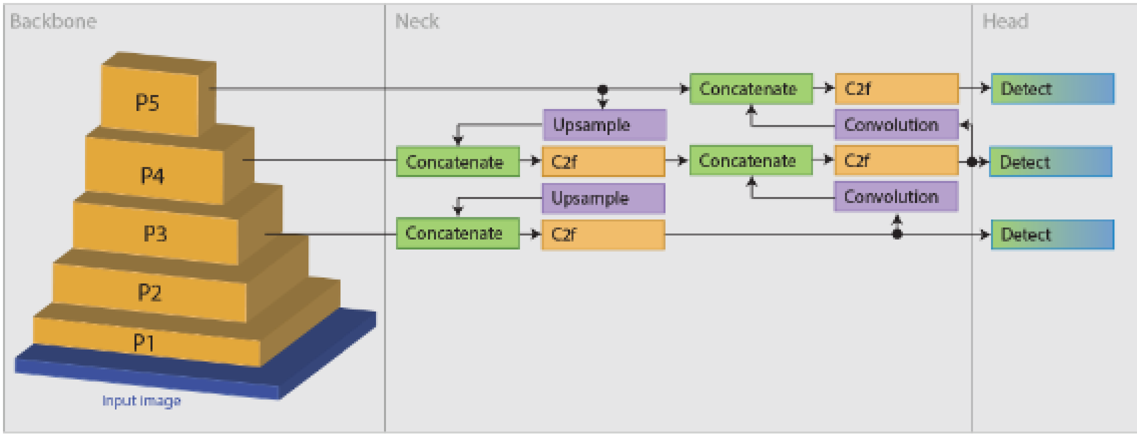

Figure 1. Schematic of the simplified architecture of the yolov8 model

The main building block that is used in the yolov8 architecture is a fast implementation of cross-stage partial bottleneck with two convolutions — denoted as C2f. This block is used in Neck without the usage of residual connections, however in Backbone it is utilized with residual connections for the bottleneck. The first element used in this block is the convolution layer, which output is tensor X_c with dimension $C \times H \times W$, where: C - the number of channels, H - the height of the image, W - the width of the image. The dimension B corresponding to the batch was omitted for simplicity. Tensor X_c is divided in half with respect to dimension C, by which two tensors of dimension $\frac{C}{2} \times H \times W$ are obtained - X_1 and X_2. Tensors X_1 goes through a Bottleneck block (2 consecutive convolutional layers with a kernel of size 3×3 and stride equal to 1×1), the number of Bottlenecks is n and for each of them, depending on the place of occurrence, a residual connection can be used. The data from each of the n Bottlenecks are concatenated with the X_2 tensor by channel dimension C. The merged tensors pass through the convolution layer which results in the output tensor Y of C2f block.

1.2. Overview of solutions used for small objects detection

Detection of very small objects in images is a topic often covered by researchers. In most cases, introduced solutions to detection model architectures are associated with an increase in model parameters, and hence the need for computing power, which increases the inference time. The solution developed by (Tong and Wu, 2024) is based on the use of two new blocks in the YOLO model architecture. These blocks are reparametrized module with channel shuffle and multi-scale feature enhanced convolution. The use of these blocks allowed to increase the mAP50 metric relative to the base model by 0.3 in the case of model size n and m. For model size s, this metric was reduced by 0.1. However, in each of these cases, the introduction of the blocks in question was associated with an increase in inference time. Similar results were obtained by (Hu *et al*., 2023), the improvement of model metrics was associated with a significant increase in the number of parameters and an increase in inference time. The one developed by (Huo *et al.*, 2021), unlike the presented research, is not based on the YOLO architecture, but uses a modified U-NET architecture for detection of very small objects. The solution achieves a faster inference time of 25FPS relative to the baseline YOLOv4 model, which achieved an inference time of 22FPS with much worse metrics. The solution presented was optimized only for the detection of small objects, not objects at different scales. It should be noted that in this publication the emphasis is not only on the detection of small objects, but on the detection of objects at various scales from very small to large, due to the need to locate aircraft both very distant and those in close proximity. For this reason, the following part of the article will analyze methods that allow the detection of small objects, as well as reduce computing power requirements due to the fact that the pilot information system must operate in real time.

Researchers (Zhai *et al.*, 2023) proposed improving the YOLO architecture with an additional branch for processing features from P2. In YOLO models, the standard size of the input image is equal to 640×640px, therefore the size of the feature maps in P_2 is 160×160px, and a single detection cell has a size of 4×4px relative to the input frame. By having a high density of feature detection cells from the P_2 level, it makes it possible to detect small objects with high precision, since each element of the feature map can represent a small part of the image. In each subsequent level, the detection cells are responsible for detecting larger and larger objects. Therefore, the architecture of yolo models using features from P_2 should do well in detecting both very small and large objects. This solution allowed to achieve micro-target identification. Features from this level were incorporated into Neck with FPN structure. Features from P2 were concatenated with up-sampled features from the lower level layers $\{P_3, P_4, P_5\}$, and thus passed on to the second step in Neck. The researchers also added an additional Head to detect micro-sized objects. The P2 layer detection Head may successfully decrease the negative impacts of scale variance when combined with the original detection head. This solution has improved the identification of small objects by using both features from P2 and adding Head, but such a solution translates into increased computing power requirements as well as memory overhead.

A solution that can allow reducing the number of parameters in Neck while maintaining or even improving the overall metrics of the model is the Asymptotic Feature Pyramid Network (AFPN) structure. This solution allows to reduce the parameters of the model and thus reduce the need for computing power and memory (Yang *et al*., 2023). The Adaptively Spatial Feature Fusion (ASFF) block can be used in the AFPN. This block is responsible for scaling tensors from other levels. In the case of up-sampling, a convolution layer with a 1x1 kernel is applied in order to compress the C dimension of the tensor, and then linear interpolation is used. For downsampling, depending on the level, a convolution layer or a convolution layer with max-pooling operation is used. The tensors are scaled to the $H \times W$ dimension from the corresponding P level from the Backbone (Liu *et al*., 2019). The principle of AFPN with ASFF block is shown in Figure 2.

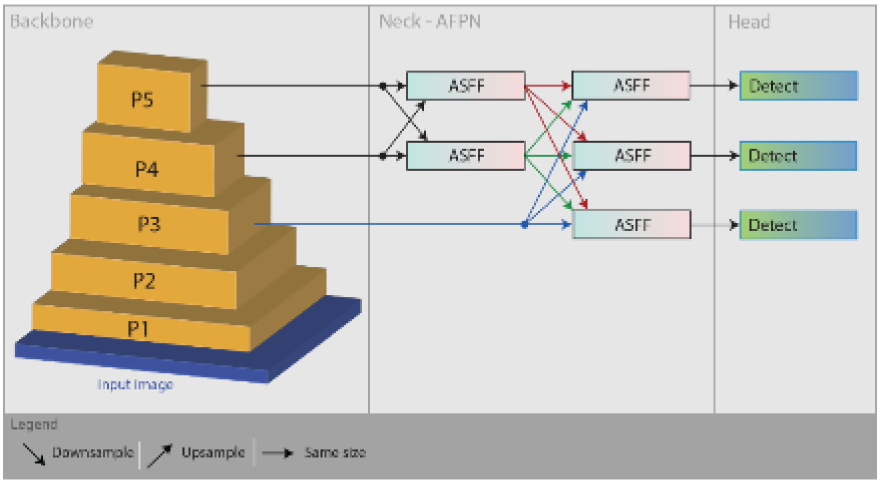

Figure 2. AFPN with ASFF block

Another noteworthy solution is the use of a preprocessing and postprocessing algorithm — Slicing Aided Hyper Inference (SAHI). This algorithm allows dividing a high-resolution image into smaller overlapping fragments (slice). Each slice is separately analyzed by the model, allowing increased efficiency for small objects. The solution also removes duplicates in the case of overlapping fragments (Akyon *et al*., 2022). A diagram of the operation of the algorithm in question is shown in Figure 3.

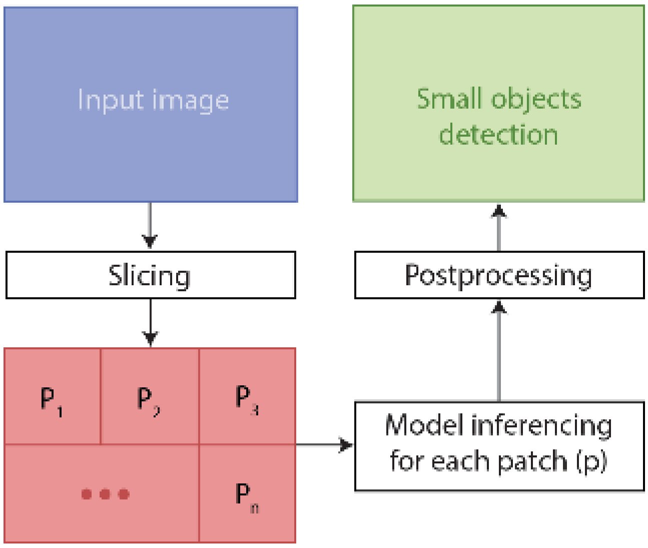

Figure 3. Schematic of the SAHI algorithm (Akyon *et al*., 2022)

2. Materials and methods

The Yolo family of model architectures for object detection was the subject of the investigation. The Yolov8 model was chosen in order to assess its efficacy and make necessary adjustments. The primary goal of the study described in the article is to create and refine the architecture so that the model can infer at a frequency of at least 25Hz and with adequate accuracy on a device like a single-board computer.

The model was examined and validated using a subset of the MS COCO 2017 (Lin *et al*., 2014) set, specifically the aircraft-containing photos with annotations. Additionally, a mosaicking procedure was used to include the AVOIDDIS (Smyers *et al*., 2023) set to the dataset. An example of the application of a mosaicking operation on an AVOIDDIS data set is shown in Figure 4.

Figure 4. An example of the application of a mosaicking operation on an AVOIDDIS data set

First of all, different sizes of the standard yolov8 and yolov8 using features from P_2 models were tested. Based on the tests, the optimal model size was determined. The models were compared with each other using metrics in the form of total number of parameters, theoretical computing power requirements expressed in Giga Floating-Point Operations Per second (GFLOPs), Average Precision (AP) metric for Intersection over Union (IoU) threshold greater than 50 and in the variable range of 50-95, F1-score, Precision, Recall and inference time and Frames Per Second (FPS). The IoU metric is defined as the quotient of the co-part of bounding boxes from the model's prediction and bounding boxes ground truth and their union. The IoU is schematically depicted in Figure 5.

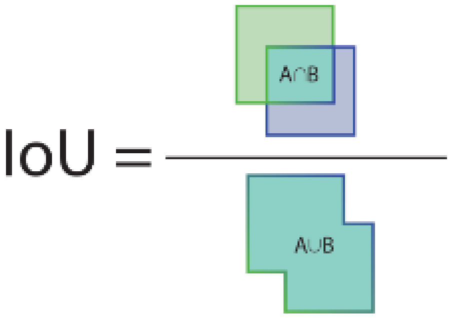

Figure 5. Schematic representation of the IoU

Both Precision M_P and Recall R_P metrics require True Positive (TP) values, which in the case of object detection occur if the IoU metric for a given bounding box is greater than 0.5, otherwise False Positive (FP) occurs. False Negative (FN) occurs when the model has not predicted a given ground truth bounding box at all. The metrics discussed are represented by Equations 1 and 2.

$$M_P = \frac{TP}{TP+FP}. \tag{1}$$

$$M_R = \frac{TP}{TP+FN}. \tag{2}$$

The F1 metric used is the harmonic mean of Precision and Recall and is defined by Equation 3.

$$M_{F1} = 2 \cdot \frac{M_P \cdot M_R}{M_P+M_R}. \tag{3}$$

The AP metric was used due to the presence of only one class in the entire data set. This metric defines the area under the Precision-Recall curve — Equation 4.

$$AP = \int_0^1 M_P(M_R). \quad (4)$$

The metrics presented were used to determine the optimal size of the model and allowed the development of an improved model architecture – 3AFPBB-YOLO. The modified version of the architecture features better metrics as well as maintaining a correspondingly small inference time.

The developed model was tested in 3 different types of tests. The first tests focused on determining the maximum distance from its model is able to detect an aircraft. The main focus of the second test was to test the effectiveness of the model to locate an aircraft when the distance is fixed. The last test was an actual test — recording the aircraft's flight from a camera installed on another aircraft. All tests were conducted on a PZL-110 Koliber 150 aircraft. A photo of the Koliber 150 aircraft is shown in Figure 5.

Figure 5. Photo of the Koliber 150 aircraft

The first 2 tests were carried out using a ground station and a device installed in the aircraft, which allows the collection of information regarding its geographical position. The data from the ground station as well as from the on-board unit are synchronized with each other using exactly the same real-time server, which allows them to be correlated. A photo of the on-board unit as well as the ground station is shown in Figure 6.

a) b)

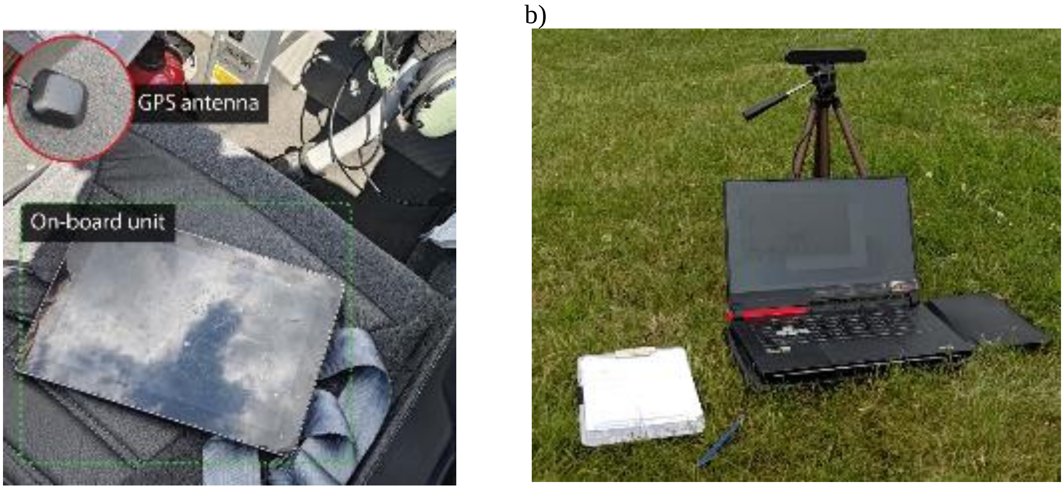

Figure 6. a) Photo of the on-board device, b) Photo of the ground station

In the third test, a video camera was installed on the PZL-104 Wilga-35A aircraft. This allowed to make a recording of the Koliber 150 aircraft in flight. The camera was installed on the wing of the aircraft. A photo of the aircraft in question with the installed camera is shown in Figure 7.

Figure 7. Photo of PZL-104 Wilga-35A aircraft with camera installed

3. Training and validation

The yolov8 models in medium (m), small (n) and nano (n) sizes, in both standard and feature versions with P_2, were trained and validated on the data set in question. The study showed that in each case the models with features from P_2 had better metrics. As expected, the best metrics were shown by the model in the largest size – medium with P_2. The model achieved an F1-score of 0.87. Models using P_2 have significantly higher theoretical computing power requirements expressed in Giga-Floating Point Operations Per second (GFLOPs) relative to the version without features from P_2, which translates into longer inference times. For example, the model with features from P_2 achieved an inference time of 10.99FPS with 97.9GFLOPs, while the model without this improvement achieved 15.08FPS with 78.7 GFLOPs. A summary of the discussed metrics obtained in the learning process is presented in Table 1.

Table 1. Comparison of model metrics obtained from the training process

Model	Parameters	GFLOPs	mAP50	mAP90	F1-score
Yolov8m	25840339	78.7	0.7683	0.5754	0.79
Yolov8m-p2	25034292	97.9	**0.8453**	**0.6494**	**0.87**
Yolov8s	11125971	28.4	0.7661	0.5469	0.79
Yolov8s-p2	10626708	36.6	0.8330	0.6150	0.86
Yolov8n	3005843	**8.1**	0.7449	0.5041	0.78
Yolov8n-p2	**2921172**	12.2	0.8150	0.5720	0.84
3AFPBB-YOLO (my)	10103441	31.3	0.8360	0.6210	0.86

During the training process, the best metrics were obtained by the Yolov8m-p2 model, which has the second largest number of parameters, and the largest theoretical computing power requirements. Models with a large number of parameters may be prone to the phenomenon of overfitting. Therefore, a more detailed analysis of the parameters will be presented on the metrics obtained from the validation set.

Tests of metrics on the validation set were also carried out, which also included determination of inference time on a cost-efficient and power-efficient Jetson Xavier NX single-board mini-computer (Jabłoński *et al*., 2022). The tests were conducted without additional optimization of the models for the hardware architecture, so the inference time results are based on PyTorch graphs (PyTorch, 2024). Table 2 shows the metrics obtained in the validation process.

Table 2. Comparison of model metrics obtained from the validation process

Model	mAP50	mAP90	Precision	Recall	Inference time [ms]*	FPS **
Yolov8m	0.826	0.681	**0.975**	0.664	62.1	15.08
Yolov8m-p2	**0.891**	**0.733**	0.973	0.791	86.8	10.99
Yolov8s	0.819	0.65	0.97	0.654	29.4	30.08
Yolov8s-p2	0.886	0.702	0.968	0.783	38.9	23.20

Continuation of Table 2

Model	mAP50	mAP90	Precision	Recall	Inference time [ms]*	FPS **
Yolov8n	0.815	0.614	0.965	0.647	**12.4**	**61.58**
Yolov8n-p2	0.875	0.661	0.954	0.769	19.6	42.99
3AFPBB-YOLO (my)	**0.891**	0.727	0.964	**0.795**	35.8	27.9

* Preprocessing and postprocessing times were not included in the inference time. The study shows that the average pre-processing time will be 1ms, while the post-processing time will be 3.2ms.
** A post- and pre-processing time of 3.2ms was added to the calculation of the average FPS.

The study shows that the optimal frame rate (~25FPS) range includes small size models, of which the model using P_2 features has much better metrics than the standard yolo model architecture. This is due to the fact that in most images the objects the model is designed to detect are quite small. For this reason, a modified architecture of the yolo model was developed - 3AFPBB-YOLO. The architecture of the developed model is based on the AFPN solution with a modified Basic Block. According to a study conducted by (Yang *et al.*, 2023), the use of AFPN translates into a reduction in the number of model parameters while maintaining or improving model performance. A detailed description of the architecture is presented in the section 3.1. Architecture, training and validation – 3AFPBB-YOLO.

The main assumption for model selection was the inference time on the Jetson Xavier NX must be greater than 25FPS while maintaining the best possible mAP50 metric. The Yolov8m, Yolov8m-p2 and Yolov8s-p2 models do not meet the basic assumption regarding inference time. The models in question are characterized by a large number of parameters and computational complexity, and therefore their inference time is longer, but they are characterized by a high value of the mAP50 metric. All tested models with the use of P_2 level features are characterized by increased inference time. Therefore, in order to reduce the inference time and to improve the metrics, a modification of the architecture was applied in the form of Neck of AFPN architecture with modified Basic Block and including features from level P_2. The combination of this solutions, allowed the mAP50 metric to achieve the same level as the Yolov8m-p2 model, with an almost 60% decrease in the number of parameters and theoretical computing power requirements. In addition, the inference time of the developed model for processing one frame is 35.8ms, while for the Yolov8m-p2 model this time is equal to 62.1ms, which translates into a 42.32% reduction in inference time. The developed solution is also characterized by the highest value of the Recall metric among all the tested solutions. Compared to models of the same size - small, the developed solution has 0.076 more mAP50 metrics than the standard yolo model with a decrease in inference time of only 6.4ms. Compared to the yolov8s-p2 model, the inference time of the developed solution is faster by 3.1ms while reducing the number of parameters by about 5%. A visual comparison of the models is shown in Figure 8. In Figure 8, the size of the markers is proportional to the number of parameters of a given model.

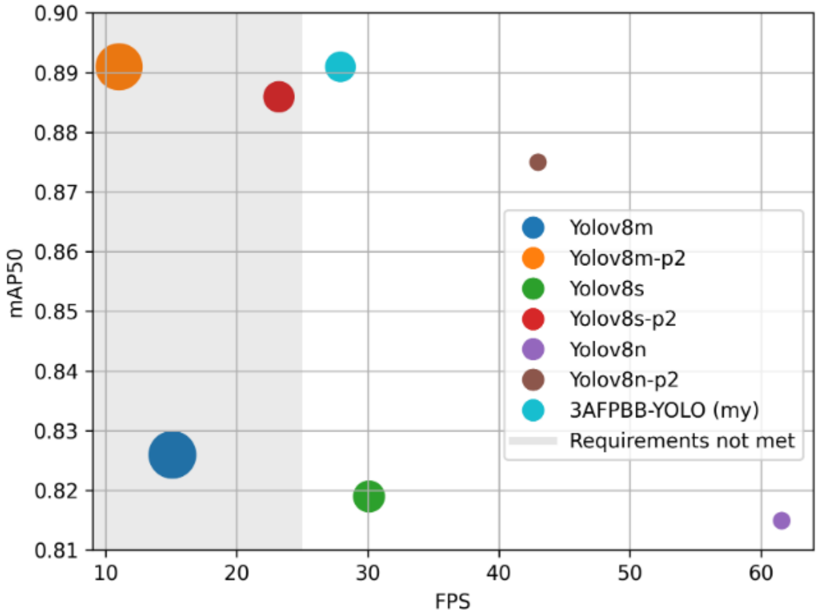

Figure 8. Visual comparison of models

3.1. Architecture, training and validation – 3AFPBB-YOLO

Due to the fact that the yolov8s-p2 model infer at a lower speed than the standard yolov8, it was decided to introduce an improvement in the form of AFPN using 3 stages. This solution is based on the use of ASFF blocks, which for the purpose of this solution were modified with Basic Block (ASFF-BB). An ASFF block in 3 stages is used, this means that in the first stage 2 ASFF blocks process data from P_2 and P_3, in the next stage features from P_4 are added, and in the last stage together all previous levels and the P_5 level features are processed. In each of these cases, new ASFF-BB blocks are added.

Basic Block is the block in which the input tensor X is the output tensor from ASFF, it is processed sequentially through the convolution layer, the batch normalization layer, the activation function in the form of SiLu, again the convolution and batch normalization layers. After which, the X tensor (residual connection) is added to the output tensor from the operation in question, and the SiLu activation function is used in the final step. A diagram of the architecture of the 3AFPBB-YOLO model including BB is shown in Figure 9.

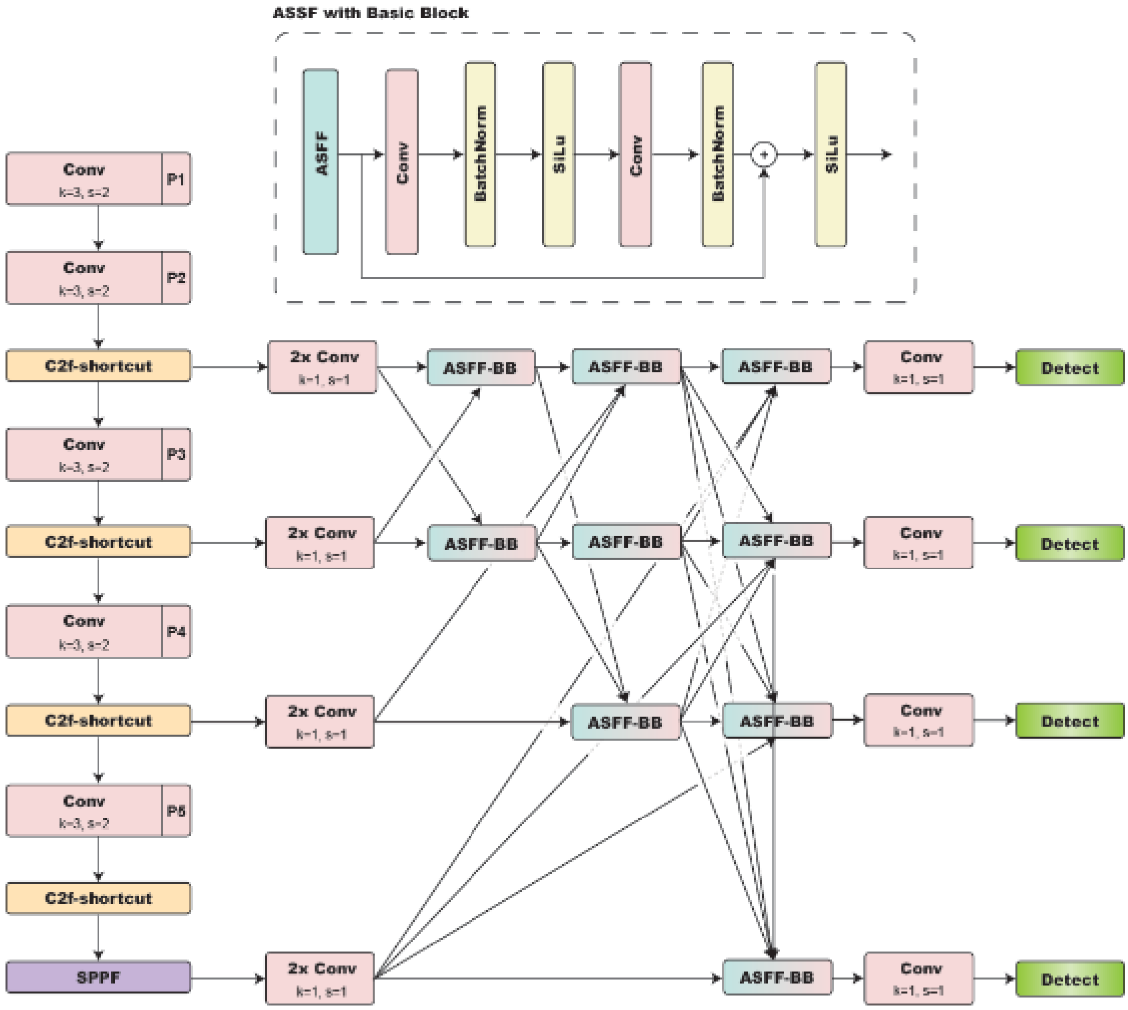

Figure 9. Simplified diagram of the architecture of the 3AFPBB-YOLO model

The developed model was trained on exactly the same data set as previous models. In the learning process, for 130 epochs the model used mosaic augmentation, and for the last 20 epochs the images were fed without mosaic augmentation. This increased the mAP50-95 from 0.563 to a level of 0.727 in case of training procedure. The exact changes in the value of the loss functions as well as the metrics per epoch are shown in Figure 10.

Based on Figure 10, it can be observed that the exclusion of mosaic augmentation for the last 20 learning epochs allowed the gradient to stabilize, which translated into a firm increase in the mAP50-95 metric, and a slight increase in the other metrics. In the case of the validation set, the value of the loss function for classes was reduced, and there was a slight decrease in the value of the box loss. Thus, the learning method used allowed, in the initial stages, the detection of multiple objects in a single image, thus increasing the diversity of the data, and thus increasing the generalization ability of the model. In the last 20 epochs, the mosaic augmentation was turned off which allowed the model to focus on precise object localization and allowed the gradient to stabilize.

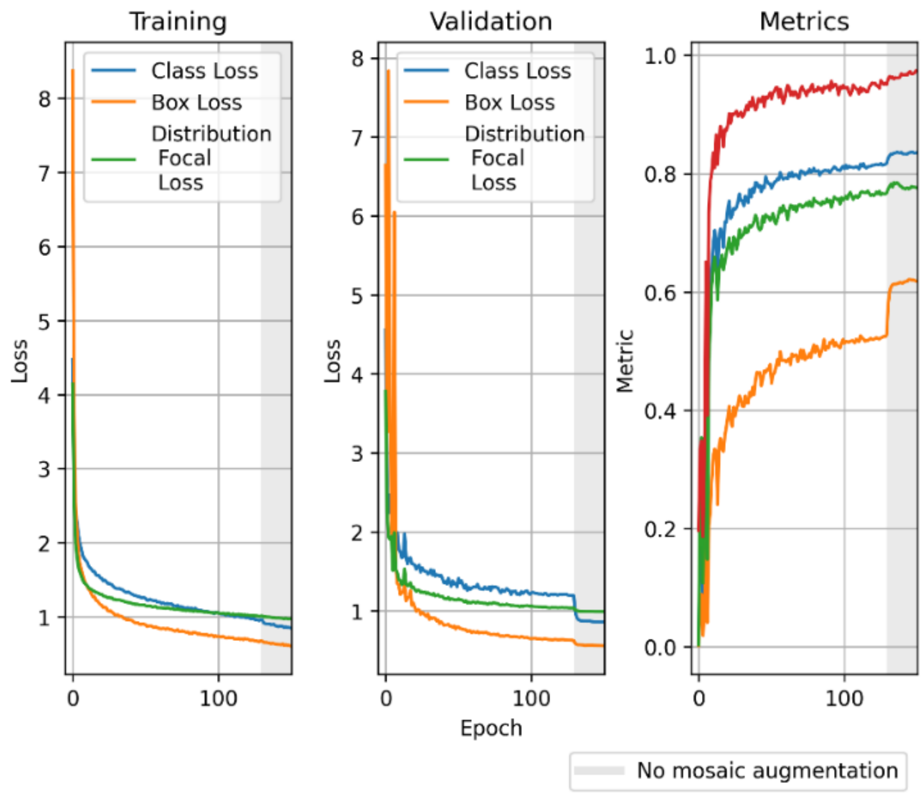

Figure 10. Changes in the value of the loss functions and the metrics per epoch

The developed model architecture has fewer parameters and GFLOPs than the architecture using the standard solution in Neck with the application of features from P_2. Above the solution allowed to obtain larger values of mAP50 and mAP90 relative to the previously presented small size models. In addition, the value of the mAP50 metric is the same as for the medium size model, and the value of the Recall metric is higher.

The system adopted a frame-by-frame analysis method, due to the fact that the aircraft detection system requires the model to detect the aircraft as quickly as possible. Analyzing a sequence of frames could reduce the complexity of the entire solution and negate the need to use other auxiliary algorithms, but this would increase the decision time, and could also translate into an increase in computing power requirements due to the need to store a sequence of frames in memory. Even in the case of analyzing short sequences of frames, the phenomena presented would take place, and in addition, hidden states would have to be stored between sequences to preserve temporal context.

The developed model analyses frames from a video camera, which are scaled to 640x640px resolution because of this, in the case of distant objects, after applying the scaling operation, information is lost that could allow identification of the aircraft. For this reason, in order to improve, a modified version of the SAHI algorithm was used, which allows inference using a batch greater than 1.

4. Results

This section presents the results obtained from the previously described tests. The effectiveness of the models with auxiliary algorithms is defined as the percentage of correct predictions E_p in the entire test. E_p is represented by Equation 5.

$$E_p = \frac{CP(F)}{F_n} \cdot 100\%, \tag{5}$$

where:

$$CP(F) = \begin{cases} 1 & \text{for correct prediction} \\ 0 & \text{for wrong or lack of prediction} \end{cases},$$

F – frame from the video;
F_n – number of frames in the video;

The haversine formula D - equation (6) was used to calculate the distance between the base station and the aircraft.

$$D(\varphi_1, \varphi_2, \lambda_1, \lambda_2) = \operatorname{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \cdot r, \tag{6}$$

where:

$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$
φ – latitude;
λ – longitude;
r – mean Earth radius [m].

The first test examined the maximum distance from which the model is able to detect an aircraft. The second test was conducted at the airport in Radawiec (Poland) using a ground station and an on-board device. The main objective of the first test was to determine the effective distance from which the model is able to return consistent predictions. The model by itself was not able to determine the location of the aircraft quite well, so it was decided to use the SAHI algorithm. Consistent predictions are defined as consecutive correct predictions where the determination of the next position does not occur later than 1s (25 frames). The first correct prediction occurred at a distance of 1030.18m, but it was not consistent due to the fact that jittering occurred and as the aircraft approached the ground station the model was unable to determine its position. The sequence of consistent prediction started from a distance of 705.09m. The values in question are shown in Figure 11, while Figure 12 shows superimposed 2 frames from the video with first prediction and first consistent prediction.

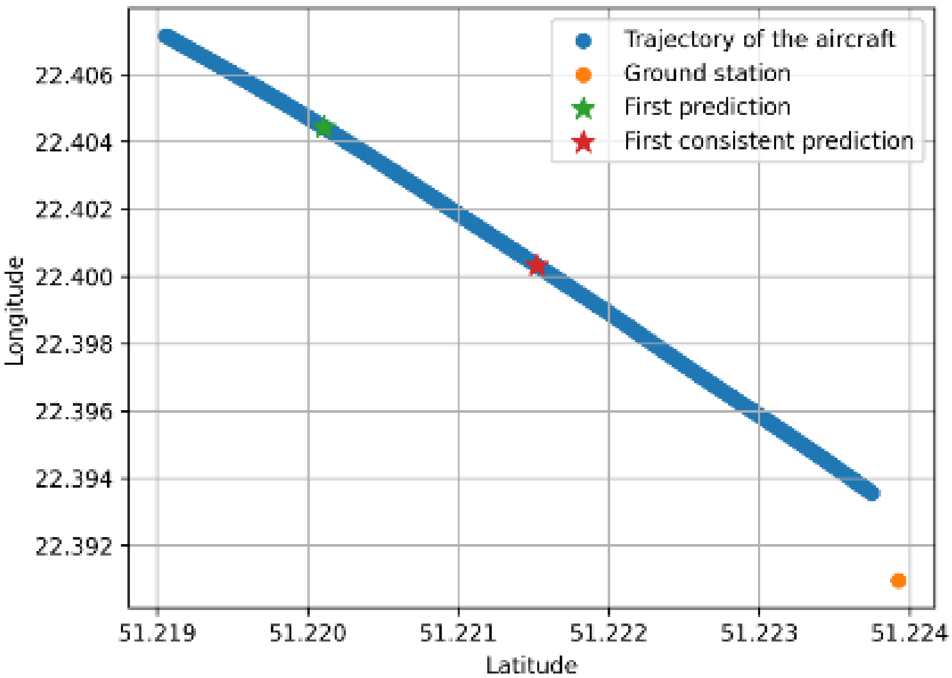

Figure 11. Visualization of geographic locations for the first test

Figure 12. Two frames from the video with first prediction and first consistent prediction

The second test examined the model's ability to locate the aircraft at a fixed distance. The aircraft was at an average distance from the ground station of 625.75 ± 17.60m (without taking into account the altitude of the aircraft). The trajectory of the aircraft as well as the location of the ground station are shown in Figure 13.

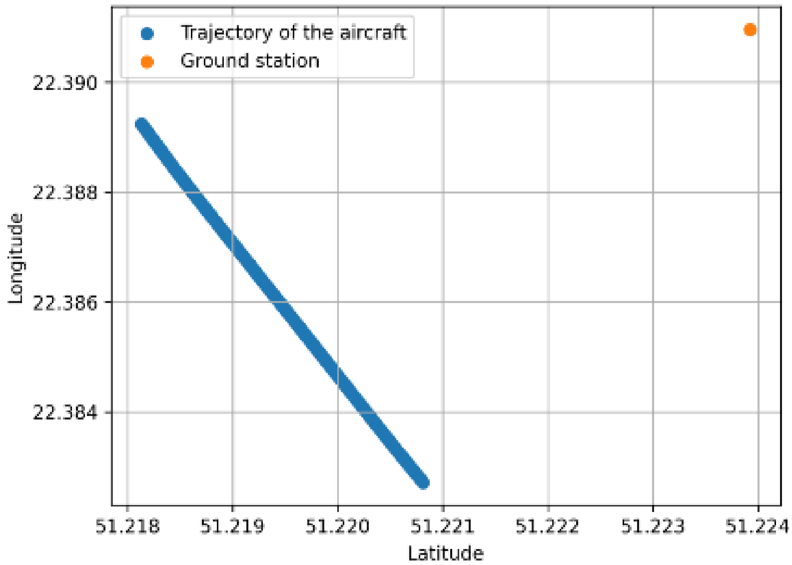

Figure 13. The trajectory of the aircraft and location of the ground station

Again, the detection performance of the algorithm alone is poor due to the small object size. The SAHI algorithm was used to improve efficiency. The model together with the auxiliary algorithm achieved an E_p efficiency of 90.14%. Examples of model prediction with the SAHI algorithm are shown in Figure 14.

Figure 14. Examples of model prediction with the SAHI algorithm

In the third test, using the developed 3AFPBB-YOLO algorithm only, a model efficiency of 79.80% was obtained on the footage taken from the Wilga 35A aircraft, which translates into 139 frames where the model failed to detect the presence of another aircraft. Example frames from the recording using the discussed model are shown in Figure 15.

Figure 15. Example frames from the recording

However, the obtained efficiency is a satisfactory result, which allows the application of additional improvements such as the use of Bounding-box Tracking with Sort (BoT-SORT) algorithm**.** The algorithm is based on the Kalman filter to predict the position of an object in subsequent frames. The Re-ID with Hungarian algorithm is also used to match objects (Aharon *et al*., 2022). By applying the tracking algorithm to the footage of a flying aircraft, the efficiency of the developed solution of 100% was achieved for this case. The algorithm detected the aircraft on every frame from the recording. An example of the model inference and the use of the BoT-SORT algorithm is shown in Figure 16.

Figure 16. Example of the model inference and the use of the BoT-SORT algorithm

5. Conclusions and future works

The 3AFPBB-YOLO artificial neural network model of small size was developed, which has better metrics than the standard yolo model and the yolo model using features from P2. The developed solution achieved mAP50 metric of 0.891, which is equivalent to the medium-sized yolo model using features from p2. This means that the proposed architecture with fewer parameters is able to achieve comparable metrics with models with more parameters. In addition, the model is able to infer in less time than the camera's sampling rate - 25FPS, which allows real-time inference.

Tests conducted showed that the model, using the SAHI algorithm, is able to detect and locate an aircraft from a distance of 1030.18m, with an effective distance of 705.09m. At a fixed distance from the ground station of 625.75 ± 17.60m, the model achieved an efficiency of 90.14%. In a test conducted using a recording of an aircraft in flight, the model by itself achieved an efficiency of 79.80%, but with the tracking algorithm, 100% efficiency was achieved.

The developed solution can be used in light aircraft, as a system that will allow to increase safety in air traffic by informing pilots about the approximate location of other aircraft. Such a solution involves the use of more than one camera, which may increase the inference time. However, the developed model has not been optimized for the device's architecture. Optimization could definitely improve the inference speed, which could allow inference on more than one camera placed on the aircraft in real time. In addition, the discussed model can serve as part of a system in autonomous aircrafts, e.g., to the coordination of multiple autonomous aircrafts.

The main limitation of the developed solution is the lack of determination of the exact distance of another aircraft, but only an indication of the direction in which it is located. However, based on the size of the bounding box determining detection on the basis of regression algorithms, it would be possible to determine the approximate distance of the aircraft - this will be the focus of future research. Further research will also focus on developing a user interface that shows the direction of where another aircraft is, and further tests of the solution will be conducted.

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.-Z. (2022) BoT-SORT: Robust associations multi-pedestrian tracking. *ArXiv,* abs/2206.14651.
2. Akyon, F.C., Altinuc, S.O., Temizel, A. (2022) Slicing aided hyper inference and fine-tuning for small object detection. In: *Proceedings of IEEE International Conference on Image Processing (ICIP),* Bordeaux, October 2022. IEEE, pp. 966-970. https://doi.org/10.1109/ICIP46576.2022.9897990.
3. Džunda, M., Dzurovčin, P., Melníková, L. (2022) Anti-collision system for small civil aircraft. *Applied Sciences*, 12(3), 1648. https://doi.org/10.3390/APP12031648.
4. Fursich, B., Bamler, R., Augustin, S., Hubers, H.W., Zhu, X.X. (2016) Towards single-pixel FMCW radar reconstruction. In: *Proceedings of 4th International Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa),* Aachen, September 2016. IEEE, pp. 95–99. https://doi.org/10.1109/COSERA.2016.7745707.
5. Hu, M., Li, Z., Yu, J., Wan, X., Tan, H., Lin, Z. (2023) Efficient-lightweight YOLO: Improving small object detection in YOLO for aerial images. *Sensors,* 23(14), 6423. https://doi.org/10.3390/S23146423.
6. Huo, Z., Yan, T., Cao, W. (2021) Fast small object detection algorithm based on feature enhancement and reconstruction. In: *Proceedings of 13th International Conference on Wireless Communications and Signal Processing (WCSP),* Changsha, October 2021. IEEE, 1-5. https://doi.org/10.1109/WCSP52459.2021.9613660.
7. Federal Aviation Administration. (2011) *Introduction to TCAS II Version 7.1.* U.S. Department of Transportation.
8. Jabłoński, B., Makowski, D., Perek, P., Nowakowski, P.N.V., Sitjes, A.P., Jakubowski, M., Gao, Y., Winter, A. (2022) Evaluation of NVIDIA Xavier NX platform for real-time image processing for plasma diagnostics. *Energies*, 15(6), 2088. https://doi.org/10.3390/EN15062088
9. Kateb, F.A., Monowar, M.M., Hamid, M.A., Ohi, A.Q., Mridha, M.F. (2021) FruitDet: Attentive feature aggregation for real-time fruit detection in orchards. *Agronomy*, 11(12), 2440. https://doi.org/10.3390/AGRONOMY11122440
10. Kulisz, M., Kłosowski, G., Rymarczyk, T., Hoła, A., Niderla, K., Sikora, J. I. (2024) The use of the multi-sequential LSTM in electrical tomography for masonry wall moisture detection. *Measurement*, 234, 114860. https://doi.org/10.1016/J.MEASUREMENT.2024.114860.
11. Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dolí, P. (2014) *Microsoft COCO: Common objects in context.* In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. *Lecture Notes in Computer Science*, 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48.
12. Liu, S., Huang, D., Wang, Y. (2019) Learning spatial fusion for single-shot object detection. 10.48550/arXiv.1911.09516.
13. Maas, J., van Gent, R., Hoekstra, J. (2020) A portable primary radar for general aviation. *PLoS One*, 15(10), e0239892. https://doi.org/10.1371/JOURNAL.PONE.0239892.
14. PyTorch (2024) *PyTorch Foundation.* Available at https://pytorch.org/ (accessed 10.18.24).
15. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B. (2021) High-resolution image synthesis with latent diffusion models. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* New Orleans, LA, USA, June 2022. IEEE, 10674-10685. https://doi.org/10.1109/CVPR52688.2022.01042.

16. Smyers, E.Q., Katz, S.M., Corso, A.L., Kochenderfer, M.J. (2023) *AVOIDDS: Aircraft vision-based intruder detection dataset and simulator*. Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track. https://doi.org/10.48550/arXiv.2306.11203.
17. Tan, M., Pang, R., Le, Q. V. (2020) EfficientDet: Scalable and efficient object detection. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, June 2020. IEEE, pp. 10778-10787. https://doi.org/10.1109/CVPR42600.2020.01079.
18. Tang, J. (2017) Review: Analysis and improvement of traffic alert and collision avoidance system. *IEEE Access*, 5, 21419–21429. https://doi.org/10.1109/ACCESS.2017.2757598.
19. Terven, J.R., Cordova-Esparza, D.M. (2023) A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction, 5*(4), 1680-1716. https://doi.org/10.3390/make5040083.
20. Tong, K., Wu, Y. (2024) I-YOLO: A novel single-stage framework for small object detection. *The Visual Computer*, 40(12). doi: 8927-8944. 10.1007/s00371-024-03284-8.
21. Yang, G., Lei, J., Zhu, Z., Cheng, S., Feng, Z., Liang, R. (2023) AFPN: Asymptotic feature pyramid network for object detection. In: *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Honolulu, October 2023. IEEE, 2184-2189. https://doi.org/10.1109/SMC53992.2023.10394415.
22. Yang, Z., Kang, X., Gong, Y., Wang, J. (2023) Aircraft trajectory prediction and aviation safety in ADS-B failure conditions based on neural network. *Scientific Reports*, 13, 19677. https://doi.org/10.1038/s41598-023-46914-2.
23. Zhai, X., Huang, Z., Li, T., Liu, H., Wang, S. (2023) YOLO-drone: An optimized YOLOv8 network for tiny UAV object detection. *Electronics*, 12(17), 3664. https://doi.org/10.3390/ELECTRONICS12173664.