I.J. Wireless and Microwave Technologies, **2026, 3, 277-289**
Published Online on June 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2026.03.18

A Three-Level Model for Detecting and Identifying Aviation Objects using Deep Learning

Olga Leshchenko*
Taras Shevchenko National University of Kyiv, 60, Volodymyrska Str., Kyiv, 01033, Ukraine
E-mail: olga.leshchenko@knu.ua
ORCID iD: https://orcid.org/0000-0002-3997-2785
*Corresponding author

Yuriy Kravchenko
Taras Shevchenko National University of Kyiv, 60, Volodymyrska Str., Kyiv, 01033, Ukraine
E-mail: yurii.kravchenko@knu.ua
ORCID iD: https://orcid.org/0000-0002-0281-4396

Oksana Herasymenko
Taras Shevchenko National University of Kyiv, 60, Volodymyrska Str., Kyiv, 01033, Ukraine
E-mail: oksgerasymenko@knu.ua
ORCID iD: https://orcid.org/0000-0001-6804-2125

Nataliia Dakhno
Taras Shevchenko National University of Kyiv, 60, Volodymyrska Str., Kyiv, 01033, Ukraine
E-mail: nataliia.dakhno@knu.ua
ORCID iD: https://orcid.org/0000-0003-3892-4543

Olexandr Makhovych
Taras Shevchenko National University of Kyiv, 60, Volodymyrska Str., Kyiv, 01033, Ukraine
E-mail: oleksandr.makhovych@knu.ua
ORCID iD: https://orcid.org/0000-0002-4684-9881

Serhii Stavytskyi
Taras Shevchenko National University of Kyiv, 60, Volodymyrska Str., Kyiv, 01033, Ukraine
E-mail: serhiistavytskyi@knu.ua
ORCID iD: https://orcid.org/0009-0002-2137-766X

Denisa Macekova
Department of Informatics, Faculty of Management Science and Informatics, University of Zilina (Slovakia)
E-mail: denisa.macekova@fri.uniza.sk
ORCID iD: https://orcid.org/0000-0001-6460-2818

Received: 02 March, 2026; Revised: 01 April, 2026; Accepted: 23 April, 2026; Published: 08 June, 2026

Abstract: This paper presents an intelligent software solution for object identification in images using deep learning models, designed for automated interpretation of monitoring results of aviation objects and infrastructure. The proposed approach addresses the growing demand for enhanced flight safety and improved efficiency of aviation operations. To meet this demand, a three-level model is proposed: Level 1 performs object detection, Level 2 provides optical character recognition (OCR) and text normalization, and Level 3 implements fuzzy matching with an object database. Based on comparative testing of detection models, YOLOv8n was selected as the core of the three-level architecture, providing an optimal balance between real-time processing speed and detection accuracy. A detailed analysis of model architectures revealed specific advantages and limitations in identifying monitoring results from image data. Training on a specialized dataset and subsequent testing confirmed the high efficiency of the proposed solution and its ability to reliably localize objects even under challenging visual conditions such as shadows, glare, and partial occlusion. The obtained results demonstrate the significant potential of the proposed intelligent solution for extending computer vision capabilities in the monitoring of aviation objects and infrastructure. The experimental results also confirm the effectiveness of the OCR and fuzzy matching modules in improving object identification accuracy under real-world conditions.

Index Terms: Object Detection, Artificial Intelligence, Deep Learning, Image Recognition, Three-Level Model, Computer Vision, Neural Networks, Yolov8n, SSD, Faster R-CNN, OCR

1. Introduction

In recent years, the rapid expansion of air traffic and the increasing complexity of aviation infrastructure have introduced significant challenges for systematic real-time object identification. However, existing approaches often rely on manual inspection or semi-automated tools, which limits their scalability and responsiveness under dynamic operational conditions. This creates a need for intelligent automated solutions capable of both accurate detection and fast decision-making. Timely detection of potentially hazardous situations and operational deviations is a crucial factor in improving the reliability and safety of aviation systems. One of the most promising directions for the development of such solutions is the application of deep learning methods for the automated processing of images obtained during aviation monitoring.

Deep learning, one of the most advanced branches of machine learning, enables the development of complex models for processing and analyzing large datasets generated by the operation of aviation systems and infrastructure. Deep learning methods enable automated extraction of meaningful patterns from large-scale visual and sensor data, which is essential for real-time aviation monitoring. This capability makes it possible to solve complex tasks that previously required substantial human effort. In the context of aviation monitoring, this includes the ability to automatically and accurately identify objects in images, track changes in the condition of infrastructure, and promptly detect potentially dangerous situations.

Modern artificial intelligence technologies, particularly deep learning, allow large volumes of data to be processed in near real-time [1], which is critical for the rapid detection of deviations in the operation of aviation systems. Deep learning algorithms [2–3] can be applied, for example, to the automated analysis of images of aerodromes and aircraft in order to detect foreign objects on runways, damage to aircraft components, or anomalous behavior of ground support equipment.

The application of deep learning in the monitoring of aviation objects and infrastructure opens new opportunities for improving the efficiency of data analysis. Deep learning models not only automate the processes of object identification and anomaly detection, but also provide a high level of accuracy, which is especially important under complex and dynamically changing operational conditions. Neural networks are capable not only of classifying the states of objects, but also of assessing probable development scenarios, thereby contributing to more informed decision-making in the field of aviation safety management.

The use of deep learning for monitoring aviation objects and infrastructure is particularly relevant given the increasing intensity of air traffic, the growing complexity of aviation infrastructure, and stricter safety requirements. Artificial intelligence technologies enable the reduction of time and resources required for data acquisition and processing, thereby enhancing the efficiency of decision-making at all stages of the monitoring process, from the acquisition of raw images and sensor signals to the generation of analytical reports and operational recommendations for aircraft operators.

The scientific contribution of this work lies in the development of a unified three-level architecture that integrates object detection, OCR, and fuzzy matching specifically adapted to aviation monitoring tasks. Unlike existing approaches, the proposed model focuses on domain-specific constraints of aviation environments, including real-time processing requirements and robustness under complex visual conditions.

2. Purpose of the Work and Key Tasks

The primary objective of this research is to develop a three-level model for the automated detection and identification of aviation objects using deep learning techniques. The proposed model is designed to ensure efficient object detection while maintaining high precision, even under challenging conditions such as shadows, glare, partial occlusion, or low image quality. Key operational requirements include high processing speed (specifically for real-time applications), versatility regarding input data formats, and scalability. Meeting these requirements necessitates the efficient allocation of computational resources and a flexible neural network architecture.

Beyond technical feasibility, this study aims to evaluate the performance of a YOLO-based system within real-world environments. The research encompasses the full development lifecycle, from problem formulation to comprehensive model testing. The system is designed for practical deployment and can serve as a foundation for commercial expansion or further academic research.

To achieve these goals, the following tasks were addressed:

- Selection of an appropriate object detection model.
- Dataset preparation and preprocessing.
- Model training and performance optimization.
- Development of the model's operational algorithm.
- Implementation and experimental validation of the model.

The relevance of this study is driven by the imperative to enhance detection methods for aviation objects and infrastructure in response to escalating air traffic density, the increasing complexity of aviation environments, and more stringent flight safety requirements. Traditional data collection and analysis methods are often resource-intensive and labor-consuming, which limits their effectiveness in rapidly changing operational conditions. In this context, the application of deep learning methods offers new opportunities for automating monitoring processes and improving the accuracy of real-time data analysis.

3. Analysis of Existing Solutions

Object detection remains a fundamental task in computer vision, with widespread adoption of models such as YOLO, SSD [4], and Faster R-CNN [5] across various application domains. Comparative studies [6,7] indicate that YOLO-based approaches provide superior real-time performance, particularly for large-scale object detection, while alternative architectures achieve higher accuracy at the cost of computational complexity. Although these approaches demonstrate strong performance in general-purpose tasks, their direct application to aviation monitoring remains limited due to stringent requirements for accuracy, processing speed, and robustness under real-time operational conditions. In particular, most existing studies do not adequately address critical challenges such as partial occlusion, varying illumination conditions, and the high variability of domain-specific objects inherent to aviation environments.

In [8-9], a YOLO-based approach is applied for real-time recognition of military equipment from unmanned aerial vehicles, demonstrating the feasibility of deploying such models in dynamic environments. Similarly, the study in [10] explores the use of YOLOv5 and YOLOv8 for automatic product identification in retail systems, highlighting the adaptability of modern detection models to domain-specific tasks. However, these works are primarily focused on either military or commercial scenarios and do not address aviation infrastructure monitoring.

The integration of computer vision into autonomous and intelligent systems is further explored in [11], where deep learning methods are applied in self-driving vehicles. In addition, evolutionary approaches to neural network optimization [12, 13] demonstrate the potential of adaptive learning mechanisms for improving model performance in complex environments.

Recent research [14–16] investigates the development of intelligent control systems, user behavior prediction models, and mathematical frameworks for wireless sensor networks. These studies emphasize the importance of combining machine learning techniques with distributed and adaptive systems, which is particularly relevant for real-time monitoring applications. Furthermore, the work in [17] focuses on reducing redundant information in digital images, contributing to more efficient preprocessing and improved performance of computer vision pipelines.

Despite the significant progress in object detection and intelligent systems, there remains a limited number of studies specifically addressing the automated detection and identification of aviation objects and airport infrastructure. Existing approaches either lack domain specificity or do not consider the constraints of real-time processing in aerial monitoring scenarios.

Therefore, the proposed research aims to bridge this gap by developing a computer vision–based approach for aviation object detection that integrates modern YOLO architectures with adaptive preprocessing and system-level optimization. This work extends existing methods by focusing on aviation-specific conditions and improving detection robustness in real-world environments.

Thus, the key limitation of existing approaches is the lack of integrated solutions tailored to aviation monitoring scenarios, which require simultaneous object detection, textual data extraction, and robust identification.

4. Selection of a Deep Learning Algorithm for the Three-level Model

For the task of detecting monitoring objects related to aviation objects and infrastructure at the first level of the three-level model, three deep learning algorithms were considered: YOLO [18-21], SSD [22], and Faster R-CNN [23]. Requirements for detection accuracy, processing speed, and computational resources were defined, since Level 1 is required to operate in a near real-time mode and to ensure stable formation of input data for the subsequent levels (OCR and Fuzzy Matching).

YOLO performs well in real-time applications due to its high computational efficiency, SSD combines acceptable speed with moderate accuracy, whereas Faster R-CNN provides the highest detection accuracy at the cost of significantly greater computational time and resource consumption. For tasks that simultaneously require high processing speed and sufficient accuracy in the detection of aviation objects, YOLO proved to be the most suitable option and was therefore selected as the base algorithm for Level 1 of the three-level architecture.

Among the main limitations of YOLO are its sensitivity to the quality of input images, reduced performance on low-power devices, and limited generalization to new object types without additional retraining. Despite these constraints, a YOLO-based system demonstrates promising results and possesses significant potential for industrial deployment. Its high processing speed is achieved by processing the entire image in a single forward pass through the neural network, dividing it into a grid in which each cell predicts B bounding boxes (x, y, w, h), a confidence score C, and class probabilities, focusing on objects whose centers fall within the respective grid cells. The confidence score reflects the quality of the predicted bounding box and serves as the basis for further filtering and for transferring results to subsequent levels of the three-level model.

For clarity, below is a table summarizing the key technical specifications of the three models (table 1).

Table 1. Evaluation methods

Criterion	YOLO (v8n)	SSD	Faster R-CNN
Architecture	Single - stage	Single-pass	Two - stage
Speed (FPS)	High (45–155 FPS)	High (30–60 FPS)	Low (5–15 FPS)
Accuracy (mAP)	Medium-high	Medium	High
Small object detection	Medium	Better than YOLO	Excellent
Model complexity	Low–medium	Medium	High
Suitable for real time	Yes	Yes	Partially/no
Resource requirements	Moderate (GPU preferred)	Moderate	High (GPU required)
Resistance to complex background	Relatively high	Medium	High

To build an ensemble of algorithms based on bagging using bootstrap, samples are generated, on each of which a classifier is trained $a_i(x)$. The resulting classifier will average the responses of all algorithms (in the case of classification, this corresponds to voting):

$$Confidence = Pr(Object) \cdot IoU_{pred}^{truth}, \tag{1}$$

where:

Pr(Object) – probability of the presence of an object in the cell,

IoU – intersection coefficient over the union between the predicted frame and the real one.

IoU is an important quality metric prediction. The closer *IoU* is up to 1 – the better the prediction framework coincides with the real one.

$$IoU = \frac{Intersection\ \ area}{Union\ area} = \frac{A \cap B}{A \cup B}. \tag{2}$$

In the *YOLO* model, the general loss function plays a key role in the learning process, allowing for the simultaneous optimization of several aspects of the detection problem. It combines three main components: coordinate losses, loss confidence, and classification losses.

$$\begin{gathered} \mathrm{L} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(x_i - \hat{x}_i\right)^2 + \left(y_i - y_i\right)^2 + \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right] + \\ + \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left(C_i - C_i\right)^2 + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c)\right)^2, \end{gathered} \tag{3}$$

where:

- S^2 – total number of grid cells
- B – number of boxes predicted per grid cell
- $c \in classes$ – set of all object categories
- λ_{coord} =?, λ_{noobj} = 0.5 = 5, λ_{noobj} = 0.5 – weight coefficients that regulate influence different components functions losses

- x_i, y_i, w_i, h_i – provided coordinate and dimension values
- 1_{ij}^{obj} – indicator availability object in the frame (equals 1 if the j-th bounding box in the i-th cell is responsible for detecting an object)
- 1_{ij}^{noobj} – indicator absence object (opposite to the previous one)
- 1_{i}^{obj} – indicator availability object in the cell
- C_i – Confidence Score. Ground Truth C_i : Defined as the IoU (Intersection over Union) between the predicted box and the real box. If no object exists, $\mathrm{C}_i = 0$
- Predicted $\hat{\mathrm{C}}_i$ – model's estimate of how likely it is that the box contains an object.
- $p_i(c)$ – the conditional class probability that the object in cell *i* belongs to class *c*.

The first component, coordinate loss, is responsible for the accuracy of predicting the spatial location of objects, i.e. the coordinates of the center, width, and height of the bounding boxes. The second component, confidence loss, determines the degree of confidence of the model in the presence of an object in a particular box, which includes an estimate of the overlap (IoU) between the predicted and true box. The third component, classification loss, measures the accuracy of predicting the class of an object among possible categories.

Thus, the general loss function in YOLO allows for a balanced optimization of the model: ensuring high-quality localization of objects, reliability in detecting their presence, and correct classification. This makes the model effective for real-time applications with high accuracy requirements.

In the YOLO model, for each cell, conditional probabilities of an object belonging to a certain class are assumed:

$$p_i(c) = \Pr\left(Class_c \middle| Object\right). \tag{4}$$

The final score for each frame is calculated as the product of this probability and the model's confidence in the presence of the object:

$$IScore(c) = \Pr\left(Class \middle| Object\right) \cdot Confidence, \tag{5}$$

which allows us to take into account both localization accuracy and the probability of correct class recognition.

5. Dataset Preparation

For model training, a dataset consisting of 280 images of monitoring objects related to aviation objects and infrastructure of various shapes and sizes was constructed. Initially, it was planned to independently collect and annotate the data, which would have required labor-intensive manual labeling of objects. However, in order to optimize time, annotated images from the Google Open Images platform were partially used [24, 25], providing access to high-quality data, including classes related to aviation. This approach made it possible to combine manual data preparation with existing resources while maintaining control over both the quality and the diversity of the dataset.

The collected images were carefully checked and filtered to ensure high dataset quality. This included verifying annotation accuracy, removing images with poor quality or incorrect annotations, and adding new images when necessary. Preparing such a dataset lays the foundation for effective model training and improves its accuracy.

Thus, thanks to the use of Google Open Images, it was possible to significantly reduce the time and effort required for data preparation, while ensuring high annotation quality – a key factor for the success of model training.

This allowed more resources to be devoted to other stages of development, such as model training, optimization, and testing, which ultimately improved the overall efficiency of the project.

The dataset was divided into training, validation, and test subsets; however, due to the limited dataset size, achieving a fully balanced distribution of samples across subsets was challenging. This may have influenced the observed discrepancy between validation and test performance.

6. Technical Implementation of the Three-level Model

At the first level of the three-level model, the model receives a “raw” input image (ranging from 300×300 to 1920×1080) and performs detection of aviation objects and infrastructure elements using YOLOv8n, which has been configured and fine-tuned on a specialized dataset with appropriate annotations. The model outputs a set of bounding boxes with coordinates (x, y, w, h) and a confidence score for each detected object. To eliminate duplicate detections and spurious responses, non-maximum suppression (NMS) is applied with a threshold of approximately 0.4, which makes it possible to retain only the most probable objects for further processing.

At the second level, each cropped fragment containing a detected aviation object is passed to the Google Cloud Vision API service, which performs optical character recognition (OCR) and returns textual descriptions extracted from markings or plates (for example, tail numbers, inventory identifiers, or service labels). The extracted text is then normalized by converting it to lowercase, removing noise (measurement units, auxiliary words, special characters), eliminating stop words, and generating several candidate variants of the complete textual description of the object. The output of the second level is a structured set of cleaned text strings suitable for robust comparison with a large database of aviation objects, even in the presence of OCR errors or non-standard spellings.

At the third level, Fuzzy Matching is performed between the normalized text and a full-text database containing information about aviation objects (identifier, type, marking, status, etc.). RapidFuzz metrics (for example, fuzz.QRatio and fuzz.tokensetratio) are used for comparison, as they allow for correct handling of word permutations, different spelling variants, and minor typographical errors. For each candidate, several closest matches from the database are selected; the results are filtered using a similarity threshold (approximately 70–80%), after which a final JSON or tabular record is generated containing the aviation object identifier, its textual description, and an integrated system confidence score.

1. Data structure configuration

To enable proper training of the YOLOv8 model [26, 27], a configuration file (config.yaml) was created. This file defines the locations of the directories containing the training and validation images, as well as the list of detection classes. In this study, only a single class "aviation object" is defined for detection. This setup allows the model to focus exclusively on a single object type, enhancing training efficiency and minimizing the risk of misclassification.

Thanks to this configuration file, the model receives all the necessary information to access the data and correctly interpret the labels, which is essential for high-quality training and for the model to operate correctly in prediction mode.

2. Visualization of Annotations

To validate the accuracy of the annotations in the dataset, a Python script was implemented. This script loads each image along with its corresponding annotation file, then draws rectangles around the detected objects based on the coordinates specified in the .txt file. Thus, the developer can visually ensure that the bounding boxes encompass the aviation monitoring objects rather than other items present in the image (Figure 1). This is a crucial step in ensuring the quality of the prepared data.

3. Detection of New Images

```python
# Відображення зображення
fig, ax = plt.subplots(1)
ax.imshow(img)

# Зчитування анотацій з файлу
with open(annotations_path, 'r') as file:
    for line in file:
        class_index, x_center, y_center, bbox_width, bbox_height = map(float, line.split())
        # Перерахунок координат анотацій
        x_center *= width
        y_center *= height
        bbox_width *= width
        bbox_height *= height
        # Обрахунок координат верхнього лівого кута
        x = x_center - bbox_width / 2
        y = y_center - bbox_height / 2

        # Створення прямокутника на основі координат анотацій
        rect = patches.Rectangle((x, y),
                                 bbox_width,
                                 bbox_height,
                                 linewidth=2,
                                 edgecolor='r',
                                 facecolor='none')
        ax.add_patch(rect)
```

Fig. 1. Annotation visualization code

To evaluate the results after training, the trained yolov8n.pt model was loaded and object detection was performed on new images that were not used during training. This allows us to assess the model's ability to generalize – that is, its ability to correctly recognize aviation objects and infrastructure elements in new conditions.

The ability to visualize detection results was implemented: when the model detects an object, it draws a bounding box around it and displays the class label along with the confidence score. This enables quick and intuitive assessment of the model's performance in practice.

4. Model Training and Evaluation

The YOLOv8 model was trained from scratch using a custom configuration file (yolov8n.yaml), which defines the model architecture. The training dataset specified in config.yaml was utilized (Figure 2). Training was carried out over 20 epochs, during which the model's predictions progressively improved.

```
val.py > ...
import os
from ultralytics import YOLO
os.environ["KMP_DUPLICATE_LIB_OK"] = "TRUE"

# Load a model
model = YOLO("yolov8n.yaml")  # build a new model from scratch

# Use the model
results = model.train(data="./yolo_train/config.yaml", epochs=20)


metrics = model.val(data="./yolo_train/config.yaml")
print(metrics)

```

Fig. 2. Model training and evaluation

The model was evaluated using the built-in .val() function, which calculates performance metrics such as precision, recall, mean average precision (mAP), and others. This enables evaluation of the model performance and supports further optimization of the architecture and hyperparameters.

Thanks to the user-friendly API of the Ultralytics library, the entire process of training and validating the model was greatly streamlined, making it easier to focus on data quality and analyzing the results.

7. Model Training and Performance Optimization

Before starting training, it is essential to select an appropriate model architecture. The structure of a three-level model is presented in Figure 3.

Fig. 3. Results of multi-object detection.

The proposed system implements a three-level model for the automated detection and identification of aviation objects. At Level 1 (Detection and Cropping), an object detection model based on the YOLO architecture is applied to input images to localize aviation objects. The detected regions are defined using bounding boxes and subsequently cropped to extract regions of interest, thereby reducing background interference. At Level 2 (Optical Character Recognition), the cropped image regions are processed using an OCR module via an external API. This stage aims to extract textual information, such as identifiers or markings associated with aviation objects. The output is a text representation of the visual content. At Level 3 (Text Fragment Matching), the extracted text is compared with entries in a predefined database using fuzzy matching techniques (e.g., RapidFuzz). This approach enables tolerance to recognition errors and partial matches. The most relevant database entry is determined based on similarity scores.

Choosing a suitable configuration can significantly improve training efficiency. Hyperparameters such as the learning rate, batch size, and number of epochs are critical for successful training. Experimenting with these parameters helps to achieve the optimal balance between learning speed and the model's ability to avoid overfitting. Employing strategies like early stopping can also prevent overtraining and conserve computational resources.

Model Testing

The algorithm of the proposed three-level model is shown in Figure 4. The diagram shows the processing sequence, which includes detection of aviation objects, cropping of regions of interest, optical text recognition, and subsequent comparison of the obtained text fragments with database records.

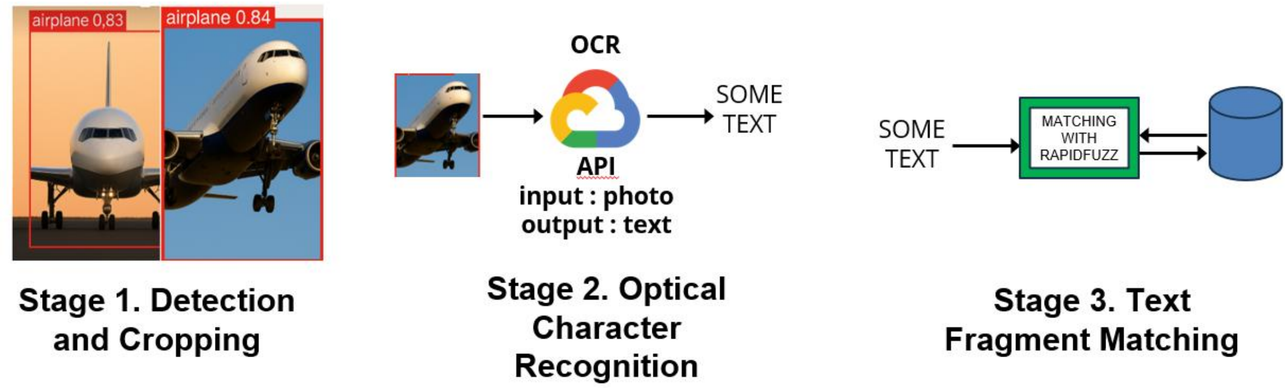

Fig. 4. Results of multi-object detection.

The model is able to accurately identify different shapes and sizes.

Evaluation and Analysis of Results

During the training process (Figure 5 and Figure 6), two series of results were obtained–one for the test set and one for the validation set.

```
fitness: 0.17683742983503417
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([    0.13829])
names: {0: 'perfume'}
plot: True
results_dict: {'metrics/precision(B)': 0.0033333333333333335, 'metrics/recall(B)': 1.0, 'metrics/mAP50(B)': 0.5237516830197195, 'metrics/mAP50-95(B)': 0.13
829140170340246, 'fitness': 0.17683742983503417}
save_dir: WindowsPath('runs/detect/train142')
speed: {'preprocess': 1.8150068976081393, 'inference': 63.12713448303463, 'loss': 8.275925085462372e-05, 'postprocess': 2.826089655643265}
task: 'detect'
PS C:\programme\open_images>
```

Fig. 5. Metrics for the test sample.

```
fitness: 0.6893824020116712
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([    0.65917])
names: {0: 'perfume'}
plot: True
results_dict: {'metrics/precision(B)': 0.9571400045280006, 'metrics/recall(B)': 0.9310344827586207, 'metrics/mAP50(B)': 0.961321005559584
3, 'metrics/mAP50-95(B)': 0.6591670016174587, 'fitness': 0.6893824020116712}
save_dir: WindowsPath('runs/detect/val5')
speed: {'preprocess': 2.3881896571577363, 'inference': 79.96218620823568, 'loss': 0.00028275876659257656, 'postprocess': 0.64461379199577
82}
task: 'detect'
```

Fig. 6. Metrics for the validation sample.

The metrics on the test dataset indicated relatively low values:

- precision ≈ 0.32
- recall = 1.0
- mAP@0.5 = 0.52
- mAP@0.5:0.95 = 0.13

Validation results were significantly better:

- precision ≈ 0.96
- recall ≈ 0.93
- mAP@0.5 ≈ 0.96
- mAP@0.5:0.95 ≈ 0.66

Such a discrepancy in performance metrics can be attributed to the following factors:

- an insufficient number of representative examples in the test set,
- some test samples containing poorly presented objects or incorrect annotations,
- imperfect balance between the training and test datasets.

Despite lower test metrics, qualitative evaluation indicates stable object detection performance on previously unseen data. To provide a more representative evaluation, the final performance metrics were calculated based on the validation dataset, which better reflects the model's generalization ability under controlled conditions. According to these results, the model demonstrated strong performance under the evaluated conditions, with Precision of approximately 0.96 and Recall of 0.93. The mAP@0.5 value reached 0.96, while mAP@0.5:0.95 was 0.66, confirming the model's effectiveness under stricter evaluation criteria. Compared to existing approaches reported in the literature, the obtained results are consistent with the expected performance of lightweight YOLO-based models, particularly in terms of real-time processing capabilities.

These results indicate that the proposed model achieves a strong balance between detection accuracy and computational efficiency, which is essential for real-time aviation monitoring applications. However, it should be noted that the evaluation is limited by the relatively small dataset size and the imbalance between training and test samples, which may affect the generalization of the results.

While the detection component (Level 1) is quantitatively evaluated, the performance of Levels 2 and 3 is assessed qualitatively, indicating a need for further quantitative validation of the full model.

Overall, the proposed three-level model provides high accuracy and processing speed, making it suitable for real-world aviation monitoring applications, such as infrastructure inspection and flight safety support systems.

To further improve model metrics and performance, the following steps are planned:

- Expanding the dataset with more diverse images, backgrounds, and viewing angles.
- Applying data augmentation techniques (e.g., brightness adjustment, rotation, scaling) to enhance generalization.
- Reviewing annotation quality in the test set, as labeling errors may affect metric reliability.
- Adjusting confidence and IoU thresholds to better reflect real-world deployment conditions.

Even with the initial YOLOv8n configuration and a limited dataset, the results show strong object detection capabilities. Further improvements can significantly boost performance without compromising visual accuracy.

Commercial Application and Integration Prospects

The developed detection module has significant potential for implementation in various commercial areas, including aviation monitoring systems, airport infrastructure inspection platforms, and flight safety support systems.

One of the key application directions is the integration of the proposed model into aviation monitoring platforms, enabling automated analysis and decision support. Another direction of development involves building recommendation systems that combine visual recognition of aviation objects with data on their technical condition and operational history.

The module can also be integrated into mobile or web applications, allowing operators or maintenance engineers to photograph aviation objects and quickly retrieve information about their characteristics, current status, and potential anomalies. This approach enhances personnel mobility and increases the efficiency of aviation infrastructure monitoring tasks. Overall, the integration of computer vision modules into aviation monitoring systems improves user interaction and supports the automation of diagnostic reporting.

8. Conclusion

The proposed three-level model for the automated detection and identification of aviation objects is implemented as a comprehensive software solution based on deep learning architectures. The system ensures robust recognition of aircraft and infrastructure elements under challenging visualization conditions, including cluttered backgrounds, partial occlusions, glare, and variable lighting. A comparative analysis of state-of-the-art detectors (YOLO, SSD, and Faster R-CNN) justified the selection of the YOLOv8n model as the base algorithm, as it provides an optimal trade-off between accuracy and processing speed for real-time applications.

To train the detection model, a dataset of 280 images of aviation objects and infrastructure was compiled. The combination of in-house annotations with data from open sources (including Google Open Images) contributed to consistent data preparation. The model was trained for 20 epochs using the YOLOv8n configuration and achieved the following validation results: Precision 0.96, Recall 0.93, mAP@0.5 = 0.96, and mAP@0.5:0.95 = 0.66. Although the test set metrics were lower, qualitative evaluation indicated stable recognition performance, including on previously unseen images.

The proposed three-level model – Level 1 (YOLOv8n-based detection), Level 2 (OCR and text normalization), and Level 3 (matching against an aviation objects database) – demonstrated effective performance under the evaluated conditions in aviation monitoring tasks. The integration of OCR and matching improves the interpretability of detection results; however, their quantitative contribution requires further investigation. The developed three-level model represents a step toward the digitalization of aviation monitoring processes and the application of computer vision methods in real-world scenarios.

The results obtained establish a foundation for future research, including multi-class object classification, the development of multimodal models (integrating visual, textual, and sensor data), and the implementation of decision support systems in the aviation industry. The proposed three-level model represents a significant step toward the automation of aviation monitoring processes and demonstrates the high potential of computer vision methods in real-world operational scenarios.

All the Declarations and Statements

Author Contribution Statement

Yuriy Kravchenko – Conceptualization, Methodology, and Supervision: Developed the research idea and overall architecture of the three-level model. Defined the study objectives, supervised the research process, and contributed to the formulation of the scientific contribution.
Olga Leshchenko – Methodology, Software, and Data Curation: Designed the system architecture, implemented the software module, and carried out dataset preparation and annotation. Conducted model training and validation, and analyzed the obtained results.
Oksana Herasymenko – Data Analysis and Validation: Performed evaluation of model performance, including metric analysis and interpretation of results. Contributed to the validation of experimental findings and preparation of visual materials.
Nataliia Dakhno – Investigation and Formal Analysis: Conducted the literature review, analyzed existing approaches, and contributed to the theoretical justification of the proposed method.
Serhii Stavytskyi – Software and Visualization: Assisted in the implementation of the detection pipeline, including visualization of annotations and testing procedures. Contributed to experimental setup and result presentation.
Oleksandr Makhovych – Writing – Review & Editing: Reviewed and edited the manuscript, improved the structure and clarity of the text, and ensured consistency with academic writing standards.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The Authors declared no conflicts of interest.

Funding Declaration

This research received no external funding.

Data Availability Statement

The data that support the findings of this study are from the corresponding author upon reasonable request.

Ethical Declarations

This study did not involve human participants or animals.

Acknowledgements

We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.

Declaration of Generative AI in Scholarly Writing

During the preparation of the manuscript, the authors carefully reviewed the content and take full responsibility for the accuracy and integrity of the manuscript.

Abbreviations

The following abbreviations are used in this manuscript:

YOLO – You Only Look Once
SSD – Single Shot MultiBox Detector
R-CNN – Region-based Convolutional Neural Network

OCR – Optical Character Recognition
mAP – Mean Average Precision
IoU – Intersection over Union
NMS – Non-Maximum Suppression
FPS – Frames Per Second
API – Application Programming Interface
GPU – Graphics Processing Unit
AI – Artificial Intelligence
CNN – Convolutional Neural Network

Appendix A\B\C…, with appendix tile

None.

References

[1] M. Farhadmanesh, A. Rashidi, and N. Marković, "General aviation aircraft identification at non-towered airports using a two-step computer vision-based approach," IEEE Access, vol. 10, pp. 48778–48791, 2022, doi: 10.1109/ACCESS.2022.3172963.

[2] V. Tetarwal, M. Kaur, and S. Kumar, "A comprehensive review on computer vision analysis of aerial data," Eng. Appl. Artif. Intell., vol. 156, p. 111206, 2025.

[3] P. Li and H. Li, "Research on FOD detection for airport runway based on YOLOv3," in Proc. 39th Chinese Control Conf. (CCC), Shenyang, China, 2020, pp. 7096–7099, doi: 10.23919/CCC50068.2020.9188724.

[4] S. Zhai, D. Shang, S. Wang, and S. Dong, "DF-SSD: An improved SSD object detection algorithm based on DenseNet and feature fusion," IEEE Access, vol. 8, pp. 24344–24357, 2020.

[5] P. Bharati and A. Pramanik, "Deep learning techniques—R-CNN to mask R-CNN: a survey," in Proc. Int. Conf. Comput. Intell. Pattern Recognit. (CIPR), 2019, pp. 657–668.

[6] Y. Shao, Z. Sun, A. Tan, and T. Yan, "Efficient three-dimensional point cloud object detection based on improved Complex-YOLO," Front. Neurorobot., vol. 17, p. 1092564, 2023.

[7] F. Yu, "YOLO, Faster R-CNN and SSD for cloud detection," Appl. Comput. Eng., vol. 37, pp. 239–247, 2024.

[8] M. Bobko and A. Sivokhin, "Method for identifying combat vehicles based on YOLO," Sci. Tech. Collection "Information Protection," no. 4, pp. 57–63, 2021.

[9] V. Petrivskyi et al., "Development of a modification of the method for constructing energy-efficient sensor networks using static and dynamic sensors," Eastern-Eur. J. Enterprise Technol., vol. 1, no. 9 (115), pp. 15–23, 2022, doi: 10.15587/1729-4061.2022.252988.

[10] T. Tran, M. D. Smith, and P. Nguyen, "Application of YOLO in automated retail shelf monitoring," ACM Trans. Intell. Syst., vol. 3, no. 1, pp. 1–18, 2023, doi: 10.1145/3581957.

[11] S. Bhaggiaraj, M. Priyadharsini, K. Karuppasamy, and R. Snegha, "Deep-learning-based self-driving cars using computer vision," in Proc. Int. Conf. Networking and Communications (ICNWC), Chennai, India, 2023, pp. 1–9, doi: 10.1109/ICNWC57852.2023.10127448.

[12] K. O. Stanley, "Efficient evolution of neural networks through complexification," 2004. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e158baaec08a54313c74ea0e1af1b72a6863406c

[13] X. Yao, "Evolving artificial neural networks," Proc. IEEE, vol. 87, no. 9, pp. 1423–1447, 1999.

[14] Y. Kravchenko, H. Dakhno, O. Leshchenko, A. Dudnik, and A. Miroshnyk, "Development of the intelligent control system of an unmanned car," CEUR Workshop Proc., vol. 3806, pp. 276–287, 2024. [Online]. Available: https://ceur-ws.org/Vol-3806/S_37_Dakhno.pdf

[15] Y. Kravchenko, O. Leshchenko, A. Dudnik, N. Dakhno, and H. Dakhno, "Machine learning systems for analyzing and predicting user behavior," in Proc. XI Int. Sci. Conf. Information Technology and Implementation (IT&I), CEUR Workshop Proc., 2024, pp. 138–152. [Online]. Available: http://ceur-ws.org/Vol-3909/Paper_11.pdf

[16] O. Kurchenko, K. Dukhnovska, O. Kovtun, A. Nikolaienko, and I. Yurchuk, "The Method of Effective Numerical Solution of the System of Equations of Thermal Conductivity," in Proc. Int. Sci. Conf. Information Technology and Implementation (IT&I Workshops), 2023, pp. 206–214.

[17] P. Prystavka, K. Dukhnovska, O. Kovtun, O. Leshchenko, O. Cholyshkina, and A. Zhultynska, "Devising information technology for determining the redundant information content of a digital image," Eastern-Eur. J. Enterprise Technol., vol. 6, no. 2 (114), pp. 59–70, 2021, doi: 10.15587/1729-4061.2021.248698.

[18] O. Trush, A. Dudnik, M. Trush, O. Leshchenko, K. Shmat, and R. Mykolaichuk, "Mask mode monitoring systems using IT technologies," in Proc. IEEE 4th Int. Conf. Advanced Trends Inf. Theory (ATIT), Dec. 2022, pp. 219–224.

[19] Ultralytics YOLO. [Online]. Available: https://github.com/ultralytics/ultralytics (accessed Mar. 23, 2026).

[20] M. K. A. Maaroof and M. S. Bouhlel, "Real-time object detection using YOLO-8 model: A drone-based approach," J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl., vol. 16, no. 1, pp. 190–204, 2025.

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934, 2020.

[22] NVIDIA, "SSD model on PyTorch Hub." [Online]. Available: https://pytorch.org/hub/nvidia_deeplearningexamples_ssd/ (accessed Mar. 2026).

[23] PyTorch, "Faster R-CNN model." [Online]. Available: https://pytorch.org/vision/main/models/faster_rcnn.html (accessed Mar. 2026).

[24] Google, "Open Images Dataset." [Online]. Available: https://storage.googleapis.com/openimages/web/index.html (accessed Mar. 2026).

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788.

[26] Y. Wen, X. Gao, L. Luo, and J. Li, "Improved YOLOv8-based target-precision detection algorithm for train-wheel-tread defects," Sensors, vol. 24, no. 11, p. 3477, 2024.

[27] T. Wu and Y. Dong, "YOLO-SE: Improved YOLOv8 for remote-sensing object detection and recognition," Appl. Sci., vol. 13, no. 24, p. 12977, 2023.

Authors' Profiles

Olga Leshchenko is currently an Associate Professor at the Department of Network and Internet Technologies, Taras Shevchenko National University of Kyiv, Ukraine. She holds a PhD in Engineering and has extensive teaching and research experience. Her research interests include information and communication networks, control systems, optimization methods, artificial intelligence, and deep learning.

Prof. Dr. Kravchenko Yurii, Head of the Department of Network and Internet Technologies, Faculty of Information Technologies, Taras Shevchenko National University of Kyiv. Current and previous research interests: Functional stability, Computer networks, cybersecurity, Artificial intelligence.

Dr. Oksana Herasymenko obtained her PhD from the Taras Shevchenko National University of Kyiv, Ukraine. She is currently an Associate Professor in the Department of Network and Internet Technologies at the same university. Her research interests include resource management in distributed computing systems, blockchain, software development, and databases.

Ph.D. N. Dakhno N.B. is currently working as an Associate Professor in the Department of Network and Internet Technologies at Taras Shevchenko National University of Kyiv, Ukraine. Their research area includes functional stability of complex systems, Artificial Intelligence, and Internet of Things.

Mr. O. Makhovych obtained his PhD in Engineering Science from Cherkasy State Technological University. He currently works as an Assistant Professor at the Department of Networking and Internet Technologies at Taras Shevchenko National University of Kyiv, Ukraine. His research area includes mathematical modeling of complex systems, computer simulation and numerical methods, modeling of objects with distributed parameters, computational approaches to engineering problems, and information and network technologies.

Dr. Serhii Stavytskyi obtained his PhD in Engineering from the Research Institute of Radio Technical Measurements at the National Space Agency of Ukraine. He is currently an associate professor in the Department of Network and Internet Technologies at the Faculty of Information Technologies, Taras Shevchenko National University of Kyiv, Ukraine.

Dr. Denisa Macekova, PhD., the lecturer of the Department of Informatics, Faculty of Management Science and Informatics, University of Zilina (Slovakia). Current and previous research interests: Machine Learning and Artificial intelligence.

How to cite this paper

Olga Leshchenko, Yuriy Kravchenko, Oksana Herasymenko, Nataliia Dakhno, Оlexandr Makhovych, Serhii Stavytskyi, Denisa Macekova, "A Three-Level Model for Detecting and Identifying Aviation Objects using Deep Learning", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.3, pp. 277-289, 2026. DOI:10.5815/ijwmt.2026.03.18