Security
of Infocommunication Systems
and Internet of Things

2024 Vol 2, No 1

https://doi.org/10.31861/sisiot2024.1.01001

Received 02 May 2024; revised 29 August 2024; accepted 30 August 2024; published 30 August 2024

Modelling the Identification and Classification of Military Air Objects Based on Machine Learning

Dmytro Uhryn[1,*] and Danylo Shved[2]

[1]Computer Science Department, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine
[2]Department of Information Systems and Networks, Lviv Polytechnic National University, Lviv, Ukraine

*Corresponding author (E-mail: d.ugryn@chnu.edu.ua)

ABSTRACT The article is devoted to the urgent problem of developing systems for intelligent identification of military aircraft based on artificial intelligence, machine learning and deep learning technologies as an important task for ensuring national security and increasing the efficiency of military operations. The necessity of such systems capable of automatically accurately recognizing and classifying aircraft in images is substantiated. Their advantages over traditional methods are highlighted: higher performance, speed, accuracy, elimination of the human factor. The critical importance of implementing innovative deep learning solutions to identify threats and increase the effectiveness of military operations is emphasised. Modern methods and tools for object recognition in visual data are analysed. The proposed method of collecting and pre-processing data for model training is described in detail, and a diagram of the key stages of developing a high-precision recognition system based on YOLOv8 is presented. The process of forming a high-quality training dataset from public sources and own aerial survey/satellite images using Roboflow for object annotation, creating subsets for training/validation/testing in the YOLO format is presented. Satisfactory results of fast recognition of military aircraft with high classification probabilities are demonstrated. A comparative analysis of the YOLOv8, R-CNN and GPT-4 models is presented, which shows the advantage of YOLOv8 in terms of forecasting accuracy and speed. The created model management system for setting hyperparameters, selecting object categories, and launching training/prediction processes is described. The results of testing the trained YOLOv8 are presented, which confirmed its high efficiency in accurately detecting targets in difficult conditions due to advanced deep learning algorithms. The optimality of YOLOv8 for solving the problem of military aircraft identification is substantiated.

KEYWORDS identification, military aircraft, artificial intelligence, machine learning, deep learning.

I. INTRODUCTION

Today, a variety of computer vision-based systems are actively and successfully used. They allow you to obtain information from digital images, snapshots and other visual input data, and to perform actions or provide recommendations based on this information. Such systems are used in a variety of industries, from energy and utilities to manufacturing and automotive, retail and healthcare. They are used for a variety of tasks, such as building a trajectory between two given destinations with the ability to avoid obstacles, and the market for their use continues to grow steadily.

In recent years, advances in computer vision algorithms, particularly those based on deep learning, have shown promising results in a variety of applications. One of the areas where deep learning and data processing technologies have significant potential is in the military sector, especially in the context of identifying military aircraft. Developing a system for intelligent identification of these assets based on deep learning is an important task that can make a significant contribution to military planning and national security.

Traditional identification methods require significant effort and time, and are prone to errors and manipulation. The introduction of systems based on deep learning and artificial intelligence automates the identification process, providing greater accuracy and speed. Algorithms trained on large amounts of data on various types of military vehicles are able to automatically recognise and classify them with high accuracy.

The deep learning-based identification model for analysing and classifying military aircraft is an innovative solution. It provides operators and managers with critical decision-making information, helping to improve resource efficiency and safety.

Aircraft identification plays an important role in ensuring national security and effective defence. An analysis of identification methods in this area identifies traditional methods, including radar systems, optical systems, identification by radio signals (IFF) and acoustic systems. Innovative approaches include deep learning, pattern recognition using artificial neural networks, satellite and drone technology, aperture synthesis radar technology, and radio and electromagnetic radiation analysis.

The combination of traditional and innovative methods allows us to create systems that provide high efficiency in detecting and classifying airborne objects.

In the field of defence and national security, the importance of artificial intelligence (AI) and machine learning (ML) technologies is growing every day, contributing to the efficiency of military operations. The task of identifying and classifying military aircraft is becoming increasingly complex due to the constant improvement of military technology. AI and ML algorithms help to analyse images with high accuracy, distinguishing the smallest details.

In military operations, reliable and rapid identification of airborne objects is extremely important. This allows military commanders to make informed decisions and ensures the effectiveness of their actions. The use of systems based on artificial intelligence and machine learning helps to identify potential threats, increases security and improves response time.

Compared to traditional human-based identification methods, automated systems using artificial intelligence and machine learning ensure consistent high performance and reduce the risks associated with the human factor.

These technologies can be useful not only in the military sector, but also in civil aviation, airspace control and other industries that require fast and accurate identification of airborne objects. This makes the development of universal and reliable identification systems an important task for society as a whole.

Analysing the needs in the military sector, it becomes clear that there is a need to create effective military aircraft identification systems based on artificial intelligence and machine learning. Such systems should be able to analyse large amounts of data quickly and accurately, which will significantly increase the effectiveness of military operations and ensure greater security.

II. LITERATURE REVIEW

Recently, technologies related to computer vision have attracted considerable research attention. Traditional methods of object detection based on manually created features and surface architectures are already outdated and do not meet current requirements.

The study [1] includes a detailed overview of deep learning-based object detection frameworks that address various problems such as occlusion, obstacles, and low resolution with different levels of modifications based on R-CNN (Region-Based Convolutional Neural Networks).

Through numerous changes, object detection algorithms have been improved in terms of speed and accuracy. Thanks to the tireless efforts of a large number of researchers, deep learning algorithms are rapidly evolving to offer improved performance in object detection. Applications such as pedestrian detection, medical imaging, robotics, self-driving cars, face recognition, and others are helping to save human resources in many industries. Works [2-4] provide a fundamental overview of object detection methods, including two classes of object detectors. The two-stage detector considers R-CNN, Fast R-CNN, and Faster R-CNN algorithms, while the one-stage detector considers YOLO v1, v2, v3, and SSD. Two-stage detectors focus on accuracy, while the main advantage of one-stage detectors is speed. The authors of this article present an improved version of YOLO called YOLO v3-Tiny and compare it with previous versions. Compared to classical approaches, modern data processing methods use advanced artificial intelligence and machine learning algorithms such as YOLOv8, R-CNN, and GPT-4. YOLOv8, the latest version of the well-known ‘You Only Look Once’ algorithm, effectively improves the speed and accuracy of real-time object detection. R-CNN (Region-based Convolutional Neural Networks) makes significant improvements in recognition accuracy by analysing specific regions of images. As for GPT-4, it not only efficiently processes and generates text, but is also capable of analysing images, which enhances the capabilities of multimodal data analysis. This feature makes GPT-4 a particularly valuable and powerful tool for analysis and identification.

III. MODELS AND METHODS

Artificial neural networks (ANNs), also known as multilayer perceptrons, are one of the most effective tools for object search and recognition [5, 6]. They mimic the natural functioning of the animal brain and are systems based on this principle. ANNs apply a series of mathematical operations to the input data x_i to obtain the output y [5-7].

$$y = \varphi\left(\sum_i (\omega_i \times x_i) + b\right), \qquad (1)$$

where ω_i − weighting coefficients are numerical parameters that determine the strength or significance of connections between neurons in an artificial neural network; x_i, y − the input data fed to the artificial neural network and the result it produces at the output; $\sum_i(\omega_i \times x_i)$ − the input signals are multiplied by the respective weighting factors. The obtained weighted results are summed up and this weighted sum is fed to the activation function, which also takes into account the thresholds (bases); b − threshold (bias) is an additional parameter in neural networks that allows you to shift the neuron's activation function along the abscissa axis. It acts as an additional offset that is added to the sum of the weighted inputs before the result is fed to the activation function; φ − is an activation function used in artificial neural networks to calculate the output of each neuron based on its weighted sum of inputs. The result of this function determines the value of the signal transmitted to the next layer of neurons. The key property of activation functions is that they are monotonic.

In order for a neural network to perform object detection and recognition tasks, it must be trained first.

A learning or training algorithm is a method or mathematical model that adjusts the parameters of a neural network by simulating the input environment. This is achieved by adjusting the weights and thresholds (biases) of the network. From an algorithmic point of view, training a neural network consists of selecting one particular model from a set of allowed models. Among the most well-known training algorithms are gradient descent and its variants, the Broyden-Fletcher-Goldfarb-Shannon family of algorithms, the Levenberg-Marquardt algorithm, and others. There are also optimisations of the gradient descent algorithm, such as adaptive moment estimation (Adam), adaptive subgradient methods for online learning and stochastic optimisation (Adgrad), and Nesterov acceleration [8-10].

After the neural network training phase comes the inference task, which is an algorithm that uses the trained neural network model to make predictions or conclusions on a test data set. This phase requires significantly less computing resources than training and can be performed on a graphics processing unit (GPU), as there is no need to calculate the error. Usually, an inference algorithm is divided into three stages: matching input data, selecting relevant features, and performing calculations to obtain a prediction.

The type of neural network training algorithm depends on the nature of the input and output data. Today, there are three main types of training: reinforcement learning, supervised learning (with a mentor), and unsupervised learning (without a mentor) [11-14].

Supervised learning is a process in which a pre-labelled data set is fed to the input of a neural network. Each instance of data is fed into the network, processed, and the result is compared to the target value, which is the desired output of the network. Then, according to a certain rule, the error is calculated, which leads to the adjustment of the weighting coefficients within the network. The rules for changing the weights are determined by the chosen learning algorithm. The training set vectors are fed to the input sequentially, the errors are calculated, and the weights are optimised for each vector until the total error for the entire training set reaches an acceptably low level. There is also a subfield of supervised learning - semi-supervised learning, in which both labelled and unlabelled data are used for training. In this case, a small labelled sample is used for the initial training of the model, which is then used to label the remaining unlabelled data. This approach can improve the accuracy of object search and recognition, provided that the input sample is small and the object classes are as related as possible [15, 16]. The supervised learning approach is widely used for image search and object recognition tasks. This is because large labelled datasets (images with labelled objects) are available for such tasks, and fully connected neural network layers are used. Fully connected layers allow the neural network to detect complex relationships between the pixels of the input image and the categories of objects to be recognised. The availability of large labelled datasets provides a sufficient number of examples for efficient training with the adjustment of network weights by minimising the error on the training set. The main modern neural network architectures used to solve object search and recognition tasks are illustrated in Fig. 1.

Convolutional neural networks (CNNs) are a type of neural network that is an advanced version of conventional fully connected neural networks. ANNs automate the process of extracting features from input data instead of doing it manually. The optimisation of the ANN architecture is based on reducing the amount of data pre-processing and the lack of full connectivity between neurons of neighbouring layers. This approach allows convolutional networks to automatically detect certain important features of recognised objects depending on their significance [17, 18].

The output features (filters) of the convolutional layer of a neural network can be calculated using the following formula [19, 20]:

$$n_{out} = \left[\frac{n_{in}+2p-k}{s}\right] + 1, \qquad (2)$$

where n_{in} – the number of input features, n_{out} – the number of output features, k – is the convolutional filter size, and s – is the step size.

There are many different algorithms for recognising objects in images. Convolutional Neural Networks (CNNs) are key to many of these computer vision models. Choosing the right model, which has a trade-off between speed and accuracy, is one of the challenges.

R-CNN (Region-based Convolutional Neural Networks) is an approach that combines a method for identifying potential regions of interest in an image with convolutional neural networks. To determine the regions, a selective search algorithm is used, which generates about

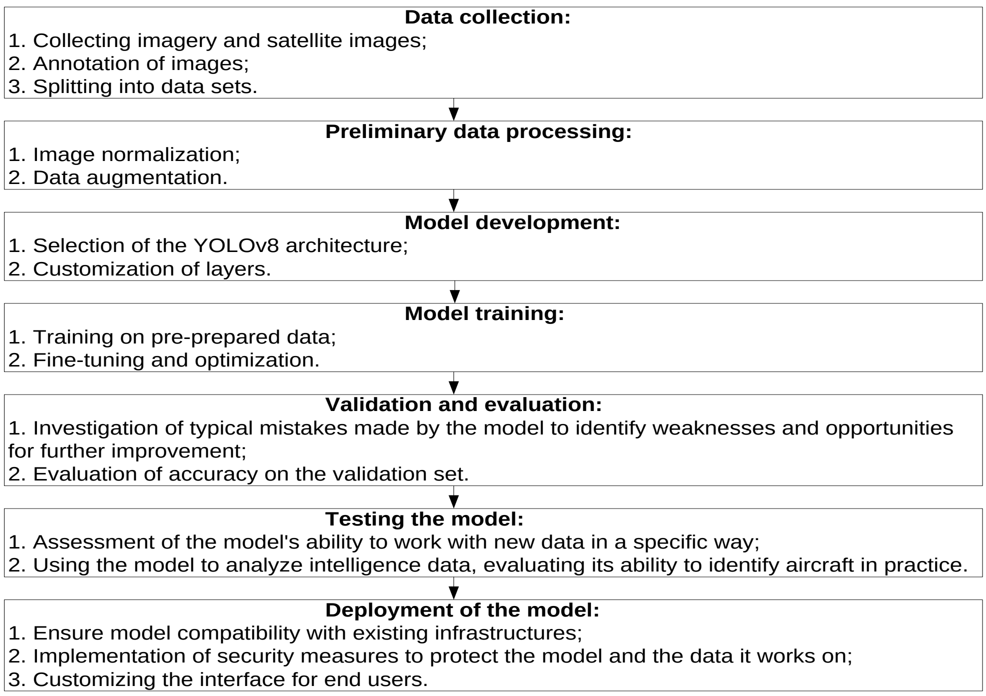


FIG. 1. Types of deep neural networks used for object recognition and detection in images.

2000 different image areas where objects are likely to be located. R-CNN helps to localise objects using a deep learning neural network and allows training high-quality models with a relatively small amount of labelled data. This method ensures high accuracy of object detection by using deep convolutional networks to classify selected regions. R-CNN allows you to scale recognition to thousands of object classes without using approximate methods such as hashing [20].

One of the solutions is YOLO (You Only Look Once), a one-step object detector designed to detect objects quickly and accurately in real time. Unlike other architectures, YOLO skips the step of selecting objects at different levels of the image, which provides higher speed.

YOLO is based on a CNN architecture that consists of multiple convolutional layers to perform convolution and fusion operations. The network also contains fully connected layers and detection layers, which are used to identify objects and their coordinates in the image.

The YOLO architecture consists of several sequential blocks that make it possible to detect objects in images or videos in real time. The main blocks include:

1. Convolutional layers to reduce image size and extract object features.
2. Maximal pooling layers for additional size reduction and increased displacement invariance.
3. Blocks with convolutional layers, batch normalisation, and activation functions for further feature extraction, stabilisation, and nonlinearity.
4. Fully connected layers for combining the results of previous layers into a vector representation.
5. Vectorization layer for converting the vector representation into a markup vector with information about the location and class of objects.
6. Layer for combining the markup vectors into one resulting vector on the image for all objects.
7. Five object detectors, each of which uses filters to predict bounding boxes and probabilities of object classes.

YOLO uses the loss of the sum of squared errors as a function, which is nevertheless important for object classification and localization tasks [21, 22]:

$$f_{lsse=}\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2]+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]$$
$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{nooobj}\left(C_i-\hat{C}_i\right)^2+\sum_{i=0}^{S^2} I_{ij}^{obj}\sum_{c\in classes}\left(p_i(c)-\hat{p}_i(c)\right)^2, \quad (3)$$

where I_{ij}^{obj} – indicates the presence of an object in a network cell.

In particular:

I_{ij}^{obj} = 1, if there is an object j in cell i (i.e., the object is detected in this cell).

I_{ij}^{obj} = 0, if there is no object j in cell i (the cell does not contain any objects).

That is, this binary (0 or 1) variable indicates whether some object j is located within cell i of the original network divided into a grid of cells. It is used to take into account the presence of objects when calculating the loss function of the object detection model.

λ_{coord} і $\lambda_{nocoord}$ – regularisation parameters are additional settings used to ensure a proper balance between different components of the loss function during neural network training. They help to avoid overfitting the model and improve its generalizability on new, unknown data.

The efficiency and accuracy of the model under study depend on the quality and variety of input data. Here we describe a method of data collection and pre-processing for training a deep learning model.

The initial step is to define data requirements. The focus is on obtaining high quality-images or satellite imagery of military aircraft from different angles, in different weather conditions and environments. This allows the model to better adapt to real-world operating conditions.

The second stage involves data collection. Both open sources (e.g., government databases, open image databases) and specialised sources (e.g., datasets from military agencies or defence companies) can be used. It is important to ensure the legal purity of data use, including copyright compliance.

The third step is to pre-process the collected data. This includes image annotation, i.e. marking objects in the photos that need to be identified. It is also important to normalise and scale the data to optimise the model training process.

The fourth step is to divide the data into sets: training, validation, and test. The training set is used to train the model, the validation set is used to adjust hyperparameters and evaluate intermediate results, and the test set is used to evaluate the overall performance of the model.

The fifth stage is continuous validation and updating of the dataset. Thus, it helps to maintain the relevance of the data and the efficiency of the model, which allows it to adapt to changes in the types and characteristics of the objects to be identified.

The sixth step involves balancing the dataset for each category of aircraft.

Fig. 2 shows a diagram of the key stages that need to be completed to create an accurate and efficient object identification system using YOLOv8 as an example.

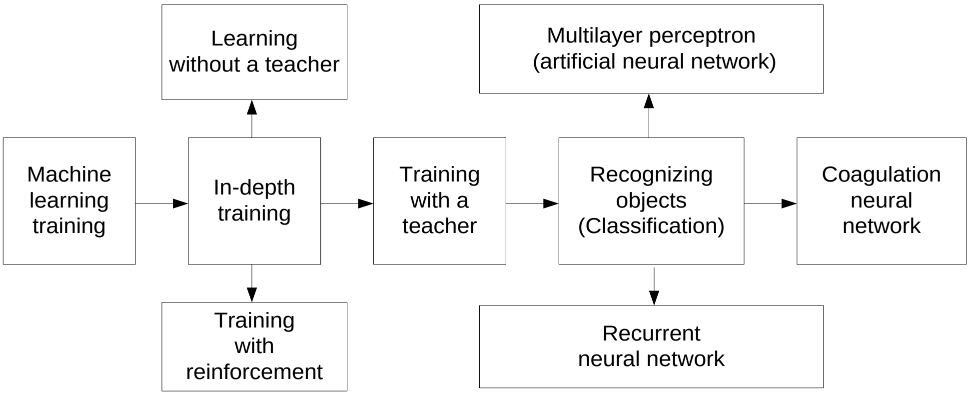


FIG. 2. Scheme of identification and classification of military aircraft objects based on the YOLOv8 model.

IV. RESULTS OF MODEL IMPLEMENTATION

To ensure a large variety and volume of data, the use of public datasets and open sources was considered. The key selection criterion was the quality and relevance of the images to the task of identifying military aircraft.

The basis for the initial dataset was a publicly available dataset from Kaggle, namely the Military Aircraft Detection Dataset [23]. This dataset includes a wide range of images of military aircrafts, which provides a large variety of data for training our model.

After selecting the main source, the next step was the process of collecting and adapting the data to the requirements of our specific task. Additional analysis of the images was performed to assess their suitability and compliance with the established criteria. The next step was to select the most representative images and annotate them for use in model training. The images to be used can be acquired using satellite technology or captured from aerial vehicles, which guarantees high resolution, as shown in Fig. 3.

The analysed data was thoroughly prepared before being used in the model training process. This preparation included image quality checks, image markup and annotation (Figures 4, 5), as well as dividing the entire dataset into sets for training, validation and testing the model.

To automatically create annotations and corresponding text files that display the coordinates of objects in the images in the dataset, we used the Roboflow resource [24].

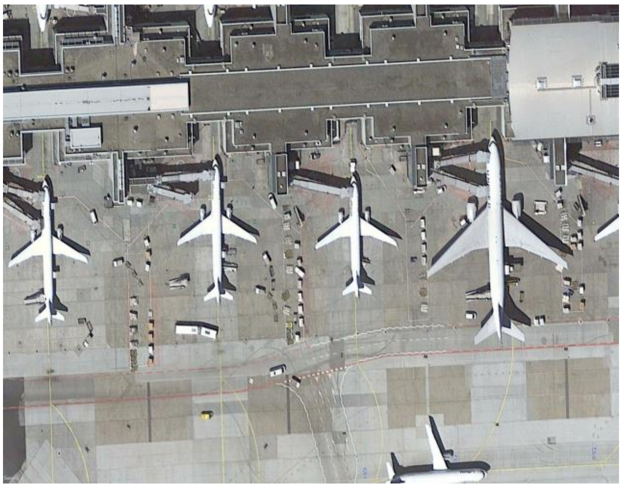


FIG. 3. Satellite image of military aircraft.

This tool helps to generate pairs of text files that correspond to the selected coordinates in the images. An example of a text file with annotated objects and their coordinates can be seen in Fig. 6.

The last step in the process of preparing a dataset is to create a dataset.yaml file. This file contains the relative paths to the images and the corresponding text files with annotations. In addition, the dataset.yaml file contains information about the number and names of classes that the network should identify during training. These class names correspond to the objects to be detected in the input images.

The next stage of the research is training the neural network. To speed up this process, we used a graphics processing unit (GPU), as it is much more efficient than a central processing unit (CPU) in the parallel computations required for neural network training. To make use of the

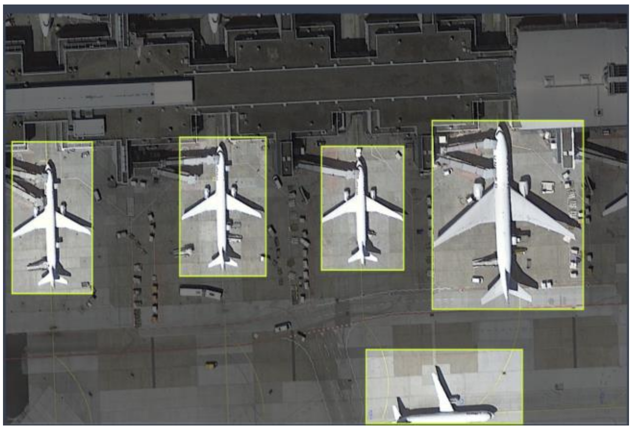


FIG. 4. Annotation of the desired element in the image.

FIG. 5. Highlighting airplane polygons in an image.

```
0 0.3471375497534021 0.37892258444591975 0.03733131190535166 0.09281036478134042
0 0.42080467191329596 0.40057833622823247 0.036335810254542245 0.10430117184950627
0 0.5727483067583067 0.12112322280877447 0.05624584327072979 0.07866783300513612
0 0.5632910410756176 0.1891841569817574 0.05126833501668282 0.0769000165331106
0 0.6612901795395776 0.6528140451488251 0.03962621086786988 0.09708234339314548
0 0.8233541095837418 0.6519461499702777 0.03369295478966228 0.09735993608319071
0 0.8941863690078327 0.6552665125016416 0.044119704458389086 0.09071871002601162
0 0.9662872544516735 0.6470193570447313 0.03506355459587751 0.0973164343915397
```

FIG. 6. Text file with coordinates of annotated objects.

GPU, a development environment was set up using CUDA, a special platform that allows you to use the power of GPUs to perform scientific computing. To start the training process using GPUs, the startup command was modified by adding a special parameter to it, which indicates that the calculations should be performed on the GPU. Detailed instructions for setting up and running the training process are shown in Figures 7 and 8.

```
aircraft_recognition_server % python train.py --img 640 --epochs 4 --data dataset.yaml --weights yolov8s.pt
```

FIG. 7. Run the workout on your computer's CPU.

```
aircraft_recognition_server % python train.py --batch 128 --data dataset.yaml --weights yolov8s.pt --device 0
```

FIG. 8. Run the workout on your computer's GPU.

```
Starting training for 5 epochs...

     Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       0/4      1.11G     0.0794    0.03523    0.02076          6        640: 100%|██████████| 340/340 [05:15<00:00,  1.08it/s]
                Class     Images  Instances          P          R      mAP50   mAP50-95:   0%|          | 0/19 [00:00<?, ?it/s]WARNING  NMS time limit 0.900s exceede
d
                Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:09<00:00,  1.94it/s]
                  all        151        242      0.313      0.502      0.355      0.131

     Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       1/4      1.11G    0.05423     0.0287    0.01346          9        640: 100%|██████████| 340/340 [04:44<00:00,  1.19it/s]
                Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:07<00:00,  2.43it/s]
                  all        151        242      0.683      0.665      0.701      0.347

     Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       2/4      1.11G    0.04903     0.0239    0.01011          6        640: 100%|██████████| 340/340 [04:50<00:00,  1.17it/s]
                Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:07<00:00,  2.51it/s]
                  all        151        242      0.717      0.708      0.741      0.382
       3/4      1.11G    0.04031    0.02165   0.008142         12        640: 100%|██████████| 340/340 [04:58<00:00,  1.14it/s]
                Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:07<00:00,  2.42it/s]
                  all        151        242      0.895      0.743      0.868       0.49

     Epoch    GPU_mem   box_loss   obj_loss   cls_loss  Instances       Size
       4/4      1.11G    0.03948    0.02157   0.006977          9        640: 100%|██████████| 340/340 [05:03<00:00,  1.12it/s]
                Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 19/19 [00:07<00:00,  2.39it/s]
                  all        151        242      0.926      0.815      0.898      0.533

5 epochs completed in 0.428 hours.
```

FIG. 9. Exploring the model with each new epoch.

When you start the learning process in the terminal, the ‘device’ parameter determines which GPU will be used for calculations. A value of ‘0’ indicates that the first available GPU will be used. Although it is possible to distribute computations between several GPUs, in this study it was decided to use only one to optimise the learning process.

The next step was to train the neural network directly. The model was trained over five epochs, and the entire process is visually represented in Figure 9

After training, the resulting models are saved in .pt format in the weights folder. The ‘best.pt’ file contains the model that showed the best results on the validation set, i.e. has the lowest error among all the models obtained during training. This model is used to recognise new images as it provides the highest accuracy.

During the study, the next step is to run the object recognition function on the selected image (Figures 10, 11, 12).

FIG. 10. The result of recognition and classification of the trained YOLOv8 model.

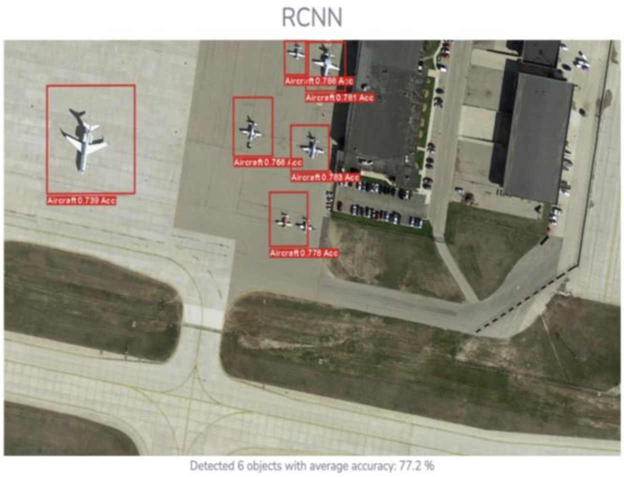


FIG. 11. Prediction result based on the R-CNN model.

FIG. 12. Prediction result of the GPT-4 model.

The numbers next to the object names indicate the probability of accurate recognition and correct classification of these objects. At the same time, other images used to test this model also demonstrated positive recognition results and were saved in the model's multimedia catalogue.

The GPT-4 model failed in the task of aircraft identification. It provided inaccurate locations of objects and indicated an incorrect accuracy rate of 90% (Fig. 12).

Based on the analysis of Fig. 10, we can conclude that the model successfully coped with the recognition of all aircraft in the image. The system identified 7 objects with an average accuracy of 95.5%.

In Fig. 11 it is clear that the R-CNN model did not successfully solve its task by failing to recognize one aircraft. In general, the prediction accuracy of the R-CNN model is significantly lower than that of YOLOv8 and amounts to 77.2%.

Based on the developed model of the military air object identification and classification system, a table of predictions for the YOLOv8, R-CNN, and GPT-4 models was created using machine learning (Fig. 13).

From this table, it can be concluded that YOLOv8 provides the most accurate predictions, and in addition, the model has a significant speed advantage over other models.

Model management (Fig. 14) allowed us to create and train our own model for classifying military aircraft.

You can manually configure hyperparameters and select the most suitable dataset categories before starting model training (Fig. 15a) using a POST request to the server. The model training process takes place in the background. To prevent possible problems with model training, validation is enabled on the training page. In addition to training the model, you can perform a run using a previously trained model (see Fig. 15b).

Predictions history

Show 10 entries Search:

Image	YOLOv8 Prediction	YOLOv8 Prediction time	YOLOv8 average acc	GPT-4 Prediction	GPT-4 Prediction time	GPT-4 average acc	RCNN Prediction	RCNN Prediction time	RCNN average acc
	Detected 6 objects	00:00:00.27	97.6%	Detected 5 objects	00:00:15.30	90.0%	Detected 6 objects	00:00:00.83	78.0%
	Detected 4 objects	00:00:00.21	94.8%	Detected 3 objects	00:00:13.18	90.0%	Detected 4 objects	00:00:00.37	77.0%
	Detected 7 objects	00:00:00.35	96.4%	Detected 5 objects	00:00:05.22	85.0%	Detected 7 objects	00:00:00.79	72.1%
	Detected 15 objects	00:00:00.83	91.8%	Detected 5 objects	00:00:25.74	90.0%	Detected 12 objects	00:00:01.49	68.7%
Image	YOLOv8 Prediction	YOLOv8 Prediction time	YOLOv8 average acc	GPT-4 Prediction	GPT-4 Prediction time	GPT-4 average acc	RCNN Prediction	RCNN Prediction time	RCNN average acc

Showing 1 to 4 of 4 entries Previous 1 Next

FIG. 13. Table of forecast history for YOLOv8, R-CNN and GPT-4 models.

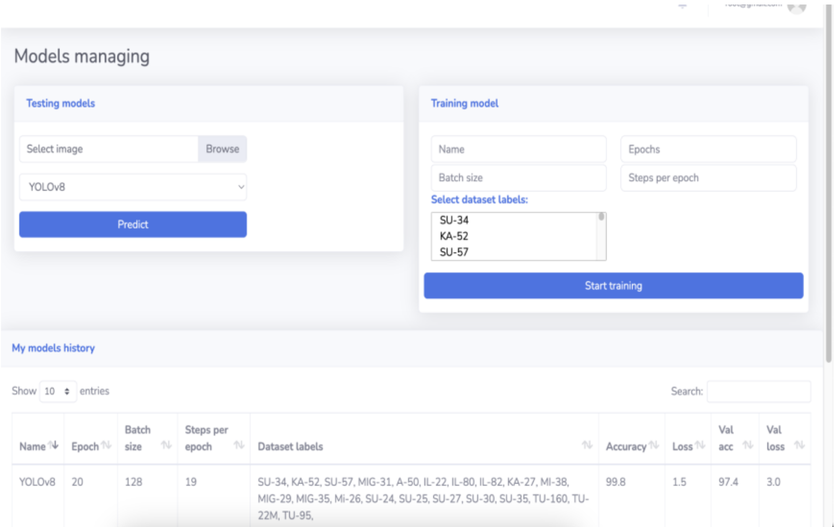
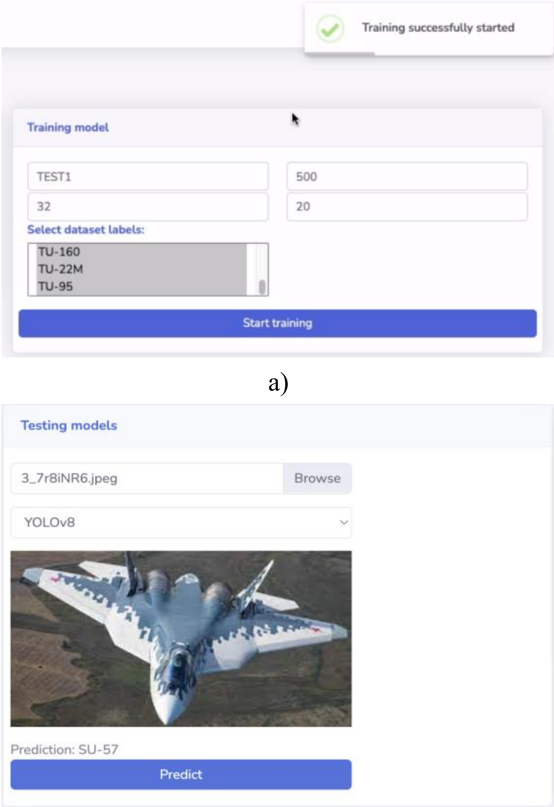


FIG. 14. Demonstration of model management.



a)

b)

FIG. 15. Displaying the training of a custom model for classifying air objects.

Based on the analysis of the classification results presented above, we can conclude that the pre-trained YOLOv8 model successfully performs the task. This model is specifically designed to accurately detect and classify objects in different environments, and it has demonstrated high accuracy in identifying target objects. One of the key factors in YOLOv8's performance is its ability to reliably recognize objects even in conditions that can be difficult for other models, such as low light, different weather conditions, or partial obscuration of an object. This is achieved thanks to the deep learning and advanced algorithms behind YOLOv8.

V. CONCLUSION

The development of intelligent identification systems for military aircraft based on artificial intelligence, machine learning, and deep learning technologies is an important task for ensuring national security and increasing the efficiency of military operations. Such systems can automatically recognize and classify different types of aircraft with high accuracy by analysing large amounts of data, including images. Compared to traditional methods of identification, AI systems provide higher performance, speed, accuracy, and eliminate the risks associated with the human factor. Implementation of innovative solutions based on deep learning is critical for increasing the effectiveness of military operations, identifying potential threats, and ensuring security in general.

Results of comparison with other studies:

1. Comparison with other deep learning models.

Studies related to object detection and classification in images have shown that different deep learning models have different levels of performance depending on the specific context. For example, studies on YOLOv3 and YOLOv4 have demonstrated significant progress in the speed and accuracy of object detection compared to previous models such as Faster R-CNN. YOLOv8, in particular, introduces improvements in the form of new architectural approaches and optimisations, making it extremely effective in the context of military aircraft. Other models, such as EfficientDet and CenterNet, have also demonstrated competitive results, but in some cases they are inferior to YOLOv8 in terms of processing speed or accuracy.

2. Comparison with traditional approaches.

Traditional methods of object identification, which include the use of manual characteristics, feature-based classification algorithms (e.g., SIFT or HOG), have limitations in comparison with modern approaches. The results show that deep learning-based systems, such as YOLOv8, outperform these methods in speed and accuracy due to their ability to automatically extract and learn from large amounts of data. Modern methods significantly reduce the dependence on human expertise and allow systems to adapt to new conditions faster.

3. Comparison with other types of neural networks.

Compared to models such as R-CNN or SSD, YOLOv8 provides better performance due to its architecture, which allows for simultaneous object detection and classification in real time. R-CNN, although it has made significant progress, often requires additional computing resources and processing time. SSD provides fast object detection, but has limitations in accuracy compared to the latest versions of YOLO.

Strengthening the comparisons with existing approaches allowed us to highlight:

1. The role of advanced deep learning algorithms.

The results demonstrating the benefits of YOLOv8 highlight the importance of new deep learning algorithms designed specifically for object detection tasks. YOLOv8 uses innovative approaches such as transformers to improve the accuracy and speed of processing. This allows it to better handle complex scenes and high noise levels in images, which is critical in a military context.

2. The impact of data quality and pre-processing.

Careful preparation of the training dataset is a key factor in achieving good results. Comparisons with other studies show that data quality and annotation directly affect model performance. The development of specialised datasets, such as public Kaggle data and satellite images, helps to improve recognition accuracy in real-world environments.

3. Real-world application.

Testing YOLOv8 under real-world conditions, such as variations in lighting and weather, demonstrates its high reliability and adaptability. Comparison with traditional methods and other state-of-the-art models highlights that YOLOv8 delivers not only accuracy but also efficiency in complex and dynamic environments.

The conclusions of this study confirm that the YOLOv8 model is extremely effective for military aircraft identification tasks due to its improved performance compared to other methods. Innovative deep learning approaches and high quality datasets play a critical role in achieving such results, which provides a significant breakthrough compared to traditional methods and existing models.

AUTHOR CONTRIBUTIONS

D.U. – conceptualization, methodology, investigation, writing-original draft preparation, visualization, supervision writing-review and editing; D.Sh. – writing-original draft preparation, software, validation, formal analysis, investigation, resources.

COMPETING INTERESTS

The authors have no competing interests.

REFERENCES

[1] Z. Zhao et al., "Object detection with deep learning: A review," *Comput. Vis. Pattern Recognit.*, vol. 1, pp. 1–21, 2019.

[2] P. Adarsh et al., "YOLO v3-Tiny. Object Detection and Recognition using one stage improved model," *6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 687–694, 2020.

[3] M. Wu et al., "Object detection based on RGC mask R-CNN," *IET Image Processing*, vol. 14, pp. 1502–1508, 2020.

[4] L. Aguilera et al., "Deep Learning CNN Implementation on Packed Malware for Cloud Cross Domain Solution Filters," *International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, pp. 192-197, 2022. [Online]. Available: https://doi.org/10.1109/ICoDSA55874.2022.9862936.

[5] G. Bonaccorso, *Mastering Machine Learning Algorithms: Expert techniques for implementing popular machine learning algorithms, fine-tuning your models, and understanding how they work*. Packt Publishing Ltd, 2020.

[6] M. Memon, "ANN vs CNN vs RNN," *Neural Networks Guide*, [Online]. Available: https://levity.ai/blog/neural-networks-cnn-ann-rnn, 2022.

[7] M. Mishra, "Convolutional Neural Networks, Explained," [Online]. Available: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939, 2020.

[8] A. Bochkovskiy et al., "AlexeyAB/darknet: YOLOv4 (version yolov4)," [Online]. Available: https://doi.org/10.5281/zenodo.562267, Zenodo, 2021.

[9] M. Elgendy, *Deep learning for vision systems (1st ed.)*. Manning, 2020. [Online]. Available: https://books.google.com.ua/books?hl=uk&lr=&id=sDszEAAAQBAJ&oi=fnd&pg=PR13&dq=Elgendy,+M.+(2020).+Deep+learning+for+vision+systems+(1st+ed.).+Manning&ots=8CCANreUy5&sig=xjoeMM3SCna2es_sEG2fI90mmXQ&redir_esc=y#v=onepage&q&f=false.

[10] S. Sah et al., "Machine learning: a review of learning types," [Online]. Available: https://doi.org/10.20944/preprints202007.0230.v1, 2020.

[11] C. Wang et al., "CSPNet: A new backbone that can enhance learning capability of CNN," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 390-391, 2020.

[12] W. Kin-Yiu, "Implementation of Scaled-YOLOv4 using PyTorch framework (v1.0.0)," *Zenodo*, [Online]. Available: https://doi.org/10.5281/zenodo.5534091, 2021.

[13] G. Jocher et al., "Ultralytics/YOLOv5: v6.0–YOLOv5n 'Nano' models. Roboflow integration. TensorFlow export, OpenCV DNN support (v6.0)," *Zenodo*, [Online]. Available: https://doi.org/10.5281/zenodo.5563715, 2021.

[14] A. Yilmaz, "Comparison of Deep Learning and Traditional Machine Learning Techniques for Classification of Pap Smear Images," pp. 63-66, 2020.

[15] N. Nguyen et al., "Semi-supervised Object Detection with Unlabeled Data," *In VISIGRAPP (5: VISAPP)*, pp. 289-296, 2019. [Online]. Available: https://doi.org/10.5220/0007345602890296.

[16] J. Kim et al., "Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition," *IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 1-4, 2020. [Online]. Available: https://doi.org/10.1109/ICCE-Asia49877.2020.9277040.

[17] A. Kuznetsova et al., "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, vol. 128(7), pp. 1956-1981, 2020. [Online]. Available: https://doi.org/10.1007/s11263-020-01316-z.

[18] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22(2), pp. 464, 2022. [Online]. Available: https://doi.org/10.3390/s22020464.

[19] X. Jiang et al., "Mnn: A universal and efficient inference engine," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 1-13, 2020.

[20] P. Puchtler and R. Peinl, "Evaluation of deep learning accelerators for object detection at the edge," *In German Conference on Artificial Intelligence (Künstliche Intelligenz)*, Springer, Cham, pp. 320-326, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-58285-2_29.

[21] N. Hoang, "Image processing-based spall object detection using Gabor filter, texture analysis, and adaptive moment estimation (Adam) optimized logistic regression models," *Advances in Civil Engineering*, 2020.

[22] M. Dorrer and A. Tolmacheva, "Comparison of the YOLOv3 and Mask R-CNN architectures' efficiency in the smart refrigerator's computer vision," *Journal of Physics: Conference Series*, vol. 1679, no. 4, pp. 42-62, 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1679/4/042022.

[23] "Military Aircraft Detection Dataset," [Online]. Available: https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset.

[24] "Roboflow," [Online]. Available: https://app.roboflow.com/.

Dmytro Uhryn

Graduated from Yuriy Fedkovych Chernivtsi National University, Chernivtsi. Currently, he is Doctor of Technical Sciences, associate professor at Yuriy Fedkovych Chernivtsi National University. He has currently published more than 170 publications. His research interests are data mining, information technologies for decision support, swarm intelligence systems, industry-specific geographic information systems.

ORCID ID: 0000-0003-4858-4511

Danylo Shved

PhD student at the Department of Information Systems and Networks of Lviv Polytechnic National University. Research interests: game theory, data science, data mining, system analysis.

ORCID ID: 0009-0005-4306-6805

Моделювання ідентифікації та класифікації військових повітряних об’єктів на основі машинного навчання

Дмитро Угрин[1,*], Данило Швед[2]

[1]Кафедра комп’ютерних наук, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна

[2]Кафедра інформаційних систем та мереж, Інститут комп'ютерних наук та інформаційних технологій, Національний університет «Львівська політехніка», Львів, Україна

*Автор-кореспондент (Електронна адреса: d.ugryn@chnu.edu.ua)

АНОТАЦІЯ Стаття присвячена актуальній проблемі розробки систем інтелектуальної ідентифікації військових повітряних засобів на основі технологій штучного інтелекту, машинного та глибинного навчання як важливого завдання для забезпечення національної безпеки та підвищення ефективності військових операцій. Обґрунтовано необхідність таких систем, здатних автоматично точно розпізнавати та класифікувати літальні апарати на зображеннях. Висвітлено їхні переваги над традиційними методами: вища продуктивність, швидкість, точність, усунення впливу людського фактора. Наголошено на критичній важливості впровадження інноваційних рішень глибинного навчання для виявлення загроз та підвищення ефективності військових дій. Проаналізовано сучасні методи та інструменти для розпізнавання об'єктів на візуальних даних. Детально описано запропонований метод збору та попередньої обробки даних для тренування моделі, наведено схему ключових етапів розробки високоточної системи розпізнавання на основі YOLOv8. Представлено процес формування якісного навчального датасету із публічних джерел та власних даних аерозйомки/супутникових знімків із застосуванням Roboflow для анотації об'єктів, створення підмножин для навчання/валідації/тестування у форматі YOLO. Продемонстровано задовільні результати швидкого розпізнавання військових літаків з високими ймовірностями класифікації. Наведено порівняльний аналіз моделей YOLOv8, R-CNN та GPT-4, що засвідчив перевагу YOLOv8 за точністю прогнозування та швидкодією. Описано створену систему керування моделями для налаштування гіперпараметрів, вибору категорій об'єктів, запуску процесів тренування/передбачення. Представлено результати тестування навченої YOLOv8, що підтвердили її високу ефективність у точному виявленні цілей за складних умов завдяки передовим алгоритмам глибинного навчання. Обґрунтовано оптимальність YOLOv8 для вирішення задачі ідентифікації військових літаків.

КЛЮЧОВІ СЛОВА ідентифікація, військові повітряні засоби, штучний інтелект, машинне навчання, глибинне навчання.