UDC 004.85:519.172:519.816 **doi: 10.32620/reks.2024.3.04**

**Viacheslav MOSKALENKO, Artem KOROBOV, Yuriy MOSKALENKO**

***Sumy State University, Sumy, Ukraine***

# OBJECT DETECTION WITH AFORDABLE ROBUSTNESS FOR UAV AERIAL IMAGERY: MODEL AND PROVIDING METHOD

*Neural network object detectors are increasingly being used for aerial video analysis, with a growing demand for onboard processing on UAVs and other limited resources. However, the vulnerability of neural networks to adversarial noise, out-of-distribution data, and fault injections reduces the functionality and reliability of these solutions. The development of detector models and training methods that simultaneously ensure computational efficiency and robustness against disturbances is an urgent scientific task.* ***The research subjects.*** *The model and method for ensuring the robustness of resource-constrained neural network systems for object detection in aerial video surveillance.* ***Objective.*** *Development of a model and method to ensure the robustness of object detectors for aerial image analysis.* ***Methods.*** *Combination of ideas and methods for dynamic neural networks, and methods for robustness and resilience optimization for neural networks.* ***Results.*** *The detector model with a ViT-B/16 backbone modified with gate units for dynamic inference was developed. The model was trained on the VEDAI dataset and meta-trained on the results of adaptation to different types of disturbances. The model with different training methods was tested for robustness against random bit-flip injection where the proportion of the modified weights is determined at a fault rate of 0.1. In addition, the model with different training methods were tested for robustness against a black-box Adversarial Attack with a perturbation level of 3/255 according to the L∞ norm.* ***Conclusions.*** *The object detection model for aerial images with dynamic inference and optimized robustness is developed for the first time. The model includes a transformer-based backbone, gate units, and simplified feature pyramid network with a RetinaNet detection head. Gate units are trained to deactivate transformer encoders that are irrelevant to the input data and disturbances. The proposed model reduces FLOPs by more than 22% without loss of mean Average Precision (mAP) by deactivating some encoders. The detector training method was developed for the first time, combined the RetinaNet loss function with the gate unit loss function and applied meta-learning to the results of adaptation to various types of synthetic disturbances. The analysis of the experimental results demonstrates that the proposed method provides an 11.7 % increase in mAP during testing under fault injection conditions and a 15.1 % increase in mAP during adversarial attack testing.*

***Keywords:*** *object detection; robustness; adversarial attack; fault injection; meta-learning.*

## 1. Introduction

### 1.1 Motivation for the research

Aerial images are characterized by significant variations in perspective distortion and scale as well as high variation in context and background [1]. This places high demands on data quality and volume and computing resources. The efficient resource utilization during aerial image processing is important for increasing the autonomy of the Unmanned Aerial Vehicles (UAV) and the speed of decision-making under resource constraints. Existing neural network compression methods to reduce resource consumption do not provide an acceptable robustness level to various kinds of disturbances acting on object recognition systems for aerial images [2].

Robustness to noise, environmental changes, and weight corruption in neural network technologies for ground object recognition is achieved by incorporating redundancy to absorb disturbances [3, 4]. Duplicate neurons or alternative neural pathways are incorporated to resist network weight corruption [4]. To enhance robustness against noise and adversarial attacks, denoising autoencoders are introduced [5], disturbance or novelty detectors are added [6], ensembling methods are employed [7], and larger neural networks are trained on data augmented with perturbations, among other techniques. There is a lack of research on effective methods to ensure robustness under resource-constrained conditions.

Object detection models have been successfully advanced toward improving architectures and microarchitectures to enhance the accuracy of localization and classification of multi-scale objects in images [8]. Architectures based on convolutional networks and vision transformers are being enhanced by implementing dynamic inference mechanisms to improve computational efficiency [9, 10]. However, most experiments are primarily focused on image classification rather than ob-

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

ject detection. However, object detectors are the most demanded in practical applications. In addition, there is a lack of research on the robustness of dynamic neural networks to various types of perturbations.

There is a research gap in terms of simultaneously ensuring robustness and reducing the computational complexity of object detection models in aerial imagery. The development of a model and method to ensure the efficient robustness of object detectors in aerial imagery by combining concepts and techniques from dynamic neural networks, robustness optimization methods, and neural network resilience is a promising area of research.

### 1.2. Objectives and Contributions

This research aims to develop a model and method for ensuring resource-efficient robustness of object detectors for aerial imagery by integrating concepts and techniques from dynamic neural networks, robustness optimization methods, and system resilience.

Robustness defines the ability of a system to withstand a certain level of stress while maintaining functionality without significant deterioration or loss of performance. In terms of robustness, robust means an object detection model that implements mechanisms to absorb and resist a certain level and type of destructive perturbations. The less the performance of the detector decreases under the influence of perturbations, the more robust is the detector. Robustness is defined as the residual functionality after exposure to an extreme destructive disturbance. We compare the robustness of the models based on their residual performance indicator after exposure to a perturbing factor.

Affordable robustness refers to an acceptable trade-off between robustness and inference time or computational complexity under the influence of perturbations. This implies the ability to adjust the trade-off between robustness and computational cost for efficient deployment on resource-constrained UAV platforms.

The key issues are as follows:

– analysis of existing solutions to ensure robustness and reduce computational complexity in object recognition systems for aerial imagery;

– development of a dynamic neural network model for object detection in aerial imagery under conditions of high observation variability and the influence of various types of noise and network weight corruption;

– development of a training method for a dynamic neural network object detector in aerial imagery to ensure robustness against noise, adversarial attacks, and neural network weight corruption;

– investigation of the dependency between the performance of the developed model and training method and certain hyperparameters of the AI system.

Structurally, the work consists of the following sections. The related works are analyzed in the Section 2. The Section 3 presents a new computationally efficient object detection model for aerial images. The Section 4 describes a new training method that provides computational efficiency and model robustness against input and weight perturbations. The Section 5 describes the experimental results of testing the proposed object detection model and training method. The research results are discussed in the Section 6. The last section concludes the paper and describes directions for future research.

## 2. The State-of-the-Art

### 2.1. Deep Learning for UAV-based Object Detection

Two-stage models (such as Faster R-CNN) and one-stage models (such as YOLO and RetinaNet) based on convolutional neural networks (CNNs) have long been the mainstream approach for object detection in images [8, 11]. These models demonstrate high accuracy in identifying and localizing objects across different scales. However, when applied to aerial imagery collected by unmanned aerial vehicles (UAVs), the architecture of convolutional networks becomes significantly more complex due to the multi-scale nature of objects and the surrounding context. This complexity increases the demand for computational resources.

Recently, Vision Transformers (ViTs) and their modifications have shown promise in providing better integration of local and global contextual information in images [12]. ViTs exhibit strong generalizability for large datasets and are particularly useful for object detection tasks that require robust performance under diverse conditions. However, despite these advantages, transformers are generally less computationally efficient than convolutional networks, which poses challenges for UAV applications where real-time processing and limited hardware are common constraints.

### 2.2. Efficient Neural Network Inference

The Bi-PAN-FPN architecture [13] improves the balance between inference speed and detection accuracy for UAV Aerial Image Recognition by integrating an improved feature pyramid network. The faster ghost module in the neck network enables a more efficient fusion of multiscale features with fewer parameters. The focus on optimizing the feature pyramid allows for enhanced performance without sacrificing computational efficiency, making it suitable for UAV applications in which detecting small targets is crucial. EUAVDet model [14] was designed to further optimize the speed-accuracy trade-off

by refining the feature fusion process. This approach involves architectural improvements that make the network lighter and faster while still being effective in challenging scenarios like small object detection.

Various model compression techniques are employed to deploy neural networks in resource-constrained environments like UAVs. Weight quantization, pruning of connections or neurons, and knowledge distillation are commonly employed to reduce the size and computational complexity of models [15, 16]. These methods maintain or improve performance while reducing the number of parameters and floating-point operations (FLOPs).

However, model compression methods can lead to a trade-off between computational efficiency and model robustness because aggressive compression may degrade accuracy and resilience in the presence of disturbances. One of the more effective approaches to enhancing adaptability and efficiency is the use of dynamic neural networks [9, 10]. Two popular strategies for dynamic neural networks are early exit networks and networks with gate units [17]. Early exit networks terminate computation early when sufficient confidence is achieved, thereby reducing computational costs for simpler inputs. Gate units, on the other hand, offer a more general approach by allowing any subset of layers to be selectively deactivated based on the input context, thereby providing significant savings in computational resources while maintaining flexibility.

### 2.3. Vulnerability and Robustness in Neural Networks

Despite advancements in architecture and efficiency, neural networks remain vulnerable to adversarial attacks, fault injections, and out-of-distribution data, which can lead to degraded performance or incorrect predictions. This vulnerability is particularly relevant for UAV systems operating in unpredictable environments. Both convolutional networks and Vision Transformers are susceptible to such disturbances [18, 19].

Recent studies have explored various techniques to improve the robustness of neural networks against such threats. In the context of UAV-based neural networks, a reactive-proactive ensemble defense mechanism is introduced to protect against adversarial attacks [20]. Although the proposed method significantly improves defense against gradient-based attacks, the use of an ensemble of models increases the computational complexity, making it less suitable for real-time applications on UAVs with strict resource constraints. Phase errors in synthetic aperture radar (SAR) imagery can be addressed by introducing the Defocusing Adaptive Complex CNN (DA-CCNN), which enhances robustness under atmospheric turbulence [21]. While their approach effectively improves target recognition accuracy, it is specific to SAR data, which limits its applicability to other types of UAV sensor data, such as visual imagery or LiDAR. Moreover, the model's complexity may hinder real-time deployment on UAV platforms where processing power is limited.

Some studies have demonstrated improvements in robustness through dynamic inference mechanisms by default [22, 23]. Additionally, fault-aware training methods have been proposed to enhance resilience against neural network weight corruption [24, 25], and adversarial training methods have been developed to counter noise and adversarial attacks [26, 27]. Other approaches focus on preprocessing input data [28], postprocessing results [29], and making architectural modifications to improve robustness [30]. The study [31] explores the use of meta-learning to improve the resilience of image classifiers against adversarial attacks.

### 3. Object Detection Model

As the backbone, a transformer ViT-B/16 or ViT-L/16 pre-trained on a large dataset was used [32]. To improve computational efficiency and adaptability to context and disturbances, we introduce gate units that dynamically switch off irrelevant or misleading transformer encoders. That is, it is assumed that only relevant features are calculated by the activated subsets of layers. The proposed architecture of the dynamic neural network backbone is illustrated in Fig. 1. Each encoder consists of a Multi-Head Self-Attention Block (MSA Block), a Multi-Layer Perceptron Block (MLP Block) with skip connectors, and a gate unit $g_k$ to activate or deactivate k-th block depending on the conditions.

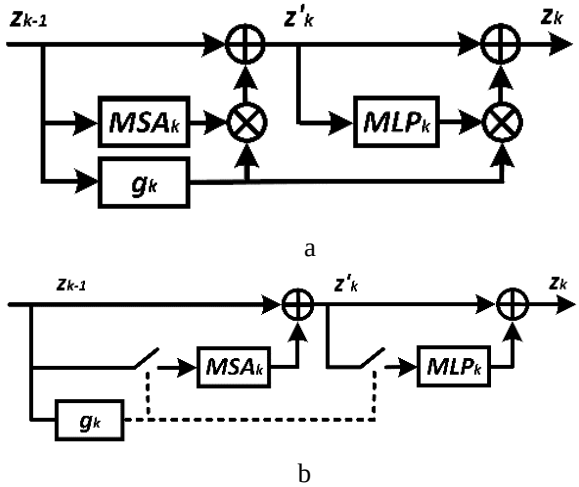


Figure 1. Schematic illustration of the structural unit of a dynamic visual transformer based backbone: a – training mode; b – inference mode

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

The dependence between the input tensor $z_{k-1}$ and the output tensor $z_k$ of the k-th block with the corresponding skip connection and gate at training time can be defined as follows:

$$z'_k = z_{k-1} + g_k(z_{k-1})\,\mathrm{MSA}_k(z_{k-1}); \quad (1)$$

$$z_k = z'_k + g_k(z_{k-1})\,\mathrm{MLP}_k(z'_k), \quad (2)$$

where $g_k$ is gate function, $g_k \in \{0,1\}$;

$f_k$ is a function of calculating the features of the k-th structural block of Visual Transformer (Multi-head Self-Attention without residual connection, Feed-Forward Network without residual connection).

The computational graph for inference can be defined as follows

$$z_k = \begin{cases} z_{k-1}, & \text{if} g_k(z_{k-1}) = 0 \\ \tilde{z}_k + \mathrm{MLP}_k(\tilde{z}_k) & \text{if } g_k(z_{k-1}) = 1 \end{cases} \quad (3)$$

where $\tilde{z}_k = \mathrm{MSA}(z_{k-1}) + z_{k-1}$.

Based on the functional purpose of the gate unit and the specifics of training multilayer neural networks, the gate unit should have the following properties [9, 10]:

- low computational complexity compared to a building block that is activated or deactivated;
- stochasticity to prevent the mode from decaying into trivial decisions, such as always or never executing a block;
- the ability to generate discrete solutions and calculate gradients to optimize the parameters of the gate unit.

Fig. 2 shows the gate unit structure in training and inference modes. The addition of Gumbel noise $G = -\log(-\log(U))$, where U ~ Unif[0, 1], to the neural output of the gate unit, allows us to add some stochasticity to avoid trivial solutions in the inference mode. The use of Gumbel-Softmax trick ensures the differentiation of the gate unit and the ability to optimize its parameters.

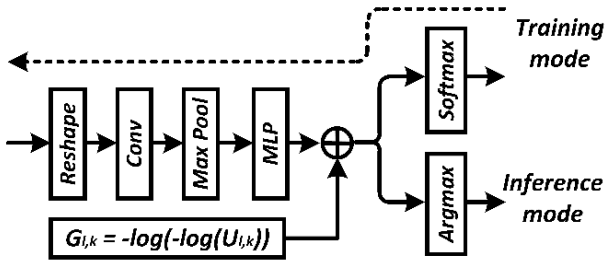


Figure 2. The architecture of the gate unit

In the experiments, the gate unit model was the same for all network blocks. MLP with one hidden layer of 24 neurons and an output layer of 2 neurons is used to calculate the relevance features of blocks in the gate unit. The activation function used is the LeakyReLU6 described above. In the case of a convolutional network, the Pooling function can be implemented as global average pooling. In the case of the visual transformer, the sequence of token vectors is first reshaped into a 2D grid similar to the intermediate representation of CNN, followed by convolution (16 filters with a 3x3 kernel) i Max Pool 2x2 [33].

Efficient adaptation of the task-agnostic plain backbone to a specific task of object detection in aerial images requires adding a specific task-specific bottleneck to the backbone output. This bottleneck should separate the most convenient features for encoding information about detected objects of different sizes. In [28], the so-called Simple Feature Piramid Network was proposed, which forms 4 different scale feature maps (Fig. 3). The 1/32 scale is constructed by stride-2 2×2 max pooling (average pooling or convolution works similarly).

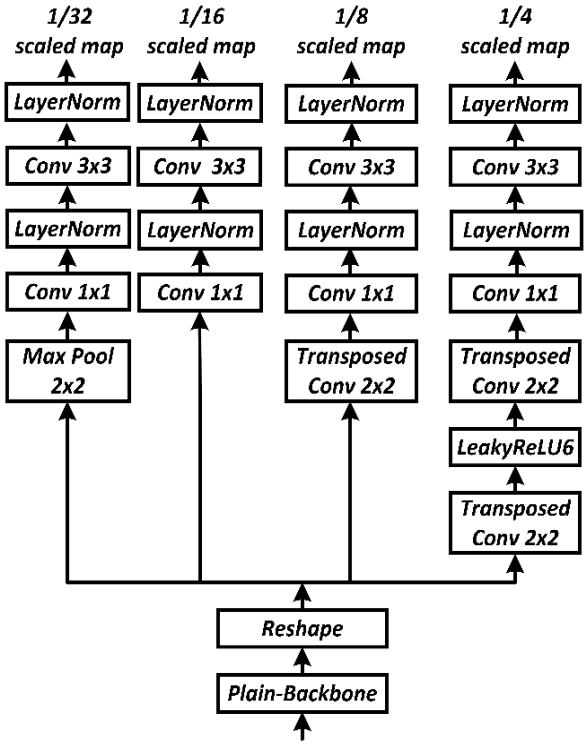


Figure 3. Architecture of Simple Feature Piramid Network for Plain Backbone

The 1/16 scale simply uses the ViT's final feature map. Scale 1/8 (or 1/4) was built by one (or two) 2×2 deconvolution layer(s) with stride=2. In the 1/4 scale case, the first deconvolution is followed by LayerNorm(LN) [34] and LeakyReLU6 [35]. Then, for each pyramid level, we

apply a 1×1 convolution with LN to reduce the dimension to 256 and then a 3×3 convolution also with LN, similar to the per-level processing of FPN.

It is proposed to use the LeakyReLU6 activation function, which combines the advantages of ReLU6 and LeakyReLU activation functions, to increase robustness against disturbances. ReLU6 reduces the attack surface by limiting the maximum value of the activation function. LeakyReLU activation function enhances network adaptation efficiency and speed by providing more informative gradients.

The detection head, which is applied to each feature map, calculates the confidence and bounding boxes for the detected objects. The detection head comprises regression box subnetworks and a classifier subnetwork (Fig. 4) [36].

It is proposed to build a one-stage detection architecture similar to RetinaNet to improve performance. 9 anchor boxes are formed for each feature map cell, each with a different size and aspect ratio [8].

Each target box is matched to anchor boxes at each training step. If the Intersection of Union (IoU) between the anchor box and target box is greater than 0.5, the corresponding anchor box is assigned to the target box. If the IoU is less than 0.4, the anchor box is considered a background box. In all other cases, the anchor box was ignored during training. The classification subnetwork is trained relative to the resulting assignments (object class or background). The regression subnetwork is trained relative to the coordinates of the selected anchor box. The error was calculated relative to the anchor box, not the target box.

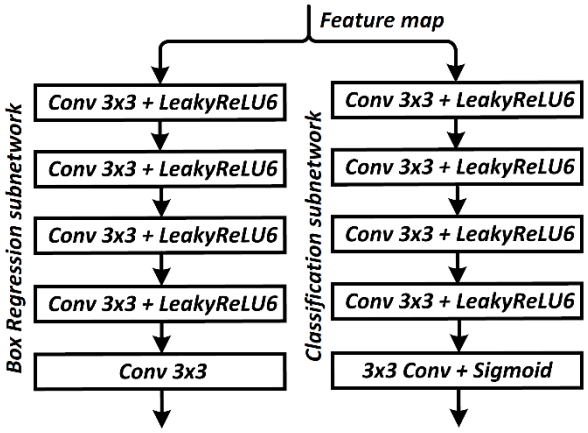


Figure 4. RetinaNet-like Detection Head

## 4. Training Method

Visual transformers can be pre-trained using any effective method. Currently, the most powerful unsupervised pretraining methods are Masked Auto-Encoding (MAE), Masked Siamese Networks (MSN) [37], Self-Distillation With No Labels (DINO) [38], and DINOv2 [39].

Fine-tuning the backbone and training the FPN with the detection head involves minimizing the composite loss function as follows:

$$L = \lambda_{loc} L_{loc} + \lambda_{cls} L_{cls} + \lambda_{usage} L_{usage}, \qquad (4)$$

where $L_{loc}$ is the regression loss of the detected objects localization on the image;

$L_{cls}$ is the classification loss of detected objects;

$L_{usage}$ is a loss that characterizes the deviation of the desired dynamic compression rate of the neural network from the real compression rate;

$\lambda_{loc}\lambda_{cls}\lambda_{usage}$ are the trade-off coefficients between the different components of the composite loss function.

The bounding box regression loss is calculated using the following formula:

$$L_{loc} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\in\{x,y,w,h\}} \text{smooth}_{L1}\left(P_{i,j} - T_{i,j}\right) \qquad (5)$$

where $P_{i,j}$ is the predicted difference between between the coordinates and the size of the anchor and target boxes;

$T_{i,j}$ are the real values of the difference in coordinates and size of the anchor and target boxes;

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 \text{ if } |x| < 1 \\ |x| - 0.5 \text{ otherwise} \end{cases} \qquad (6)$$

The classification loss is calculated using the following focal loss function formula

$$L_{cls} = -(1-p_t)^{\gamma}\log(p_t) \qquad (7)$$

where $p_t$ is the probability of predicting the i-th class;

$\gamma$ is a focus parameter.

Function (7) represents improved cross-entropy function. The difference lies in the addition of the parameter $\gamma \in (0, +\infty)$, that solves the problem of unbalanced classes. In training, most of the objects processed by the classifier are background objects, which are a separate class. Therefore, there may be a problem when the neural network learns to detect the background more effectively than other objects. The additional parameter solves this problem by reducing the error value of easily classified object classes.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

The $L_{usage}$ function that calculates the deviation of the desired dynamic compression rate of the neural network from the real compression rate. This function is affected by the number of activated blocks in the main model based on the decisions made by the gate units. The $L_{usage}$ function is calculated similarly to [9, 10],

$$L_{usage} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{n}\sum_{i=1}^{n} g_{k,i} - t_k\right)^2 \qquad (8)$$

where $g_{k,i}$ is the output of the k-th gate for the i-th training data instance;

$t_k$ is the approximated target execution rate of each neural network block in a data mini-batch, $t_k \in (0,1)$ ( $t_k = 0.5$ by default);

K is the number of gate units that control the activation of K neural network blocks;

n is the size of the training mini-batch.

To increase robustness against noise and weight faults, the detector training algorithm is based on the following principles:

- simultaneous training of the main network weights and the weights of the gate units;
- training is performed first on the main training set under normal conditions and then under episodic few-shot learning tasks of adaptation to each type of synthetic perturbation;
- generate synthetic perturbations of data or weights according to the white box scenario;
- generalization of experience during the adaptation to perturbations should be based on meta-updating using the MAML, REPTIL algorithm or meta-free weight averaging.

Ensuring the robustness of the object detection network under conditions of constrained resources is to optimize the main network and gate models so that only the components of the neural network that can provide the most accurate forecast under the influence of disturbances are calculated in the inference mode.

Let $\tau_i \mid \left\{i = \overline{1,N}\right\}$ is set of disturbance implementations relevant to the object detection system for aerial imagery [34]. Disturbances $\tau_i$ can be considered as adversarial attacks, fault injections, or switching tasks. Let $D_{base} = \left\{D_{base}^{tr}; D_{base}^{val}\right\}$ is a dataset on which the model was trained to perform the main task under known conditions. It is also given a dataset $D = \left\{D_k^{tr}; D_k^{val} \mid k = \overline{1,K}\right\}$ for K few-shot learning tasks, where fine-tuning data $D_k^{tr}$ is used in the fine-tuning stage and validation set $D_k^{val}$ is used in the meta-update stage. There is also a given set of parameters $\Psi = \langle \Theta, \phi \rangle$, and W, where $\Theta$ are parameters of the base AI model backbone, $\phi$ are parameters of gate units, and W are task specified parameters (model head parameters). Head weights $W_{base}$ for the main task should be trained on $D_{base}$.

Gradient-based meta-learning requires finding such values of the parameters $\Psi^*$, that will ensure the minimum expected loss function L on the set of implementations of different types of disturbances $\tau_i$ during its adaptation on $D_{\tau i}$

$$\Psi^* = \arg \min_{\Psi} E_{\tau \sim p(\tau)} \left[L_{\tau_i}\left(U_{\tau_i}\left(\Psi, W_{\tau_i}, D_{\tau_i}\right)\right)\right], \qquad (9)$$

where U is operator that combine a disturbing influence and adaptation in T steps and maps the current state of $\Psi$ to new state of $\Psi$.

The pseudo-code of the meta-learning algorithm for increasing the resilience of AIS is shown in Fig. 5. The stochastic gradient descent (SGD) algorithm with T steps in the U operator performs gradient meta-update $\Psi$. To simplify computation and increase stability, meta-tunable parameters can be updated using the REPTIL algorithm [41]. The type of disruptive influence does not change within a single meta-adaptation step. However, each meta-adaptation step begins with the selection of a disruptive influence type, followed by the generation of n implementations of the disruptive influence with a subsequent nested adaptation loop for each..

As shown in Fig. 5, the formation of adversarial samples is implemented by the function Adversarial_perturbation (). It is proposed to use white box attacks for meta-learning, for example FGSM attacks or PGD attacks can be used [30]. Black box attacks have been proposed for testing, for example, attacks based on the search algorithm of the covariance matrix adaptation evolution strategy (CMA-ES) [42]. The perturbation level is limited by the $L_\infty$-norm or $L_0$-norm. In this case, if the image is normalized by dividing the pixel brightness by 255, then the specified disturbance level is also divided by 255.

The Fault_injection() function generates fault injections to affect the neural network tensors [43]. It is suggested to choose the most difficult fault type to absorb, which involves generating an inversion of a randomly selected bit (bit-flip injection) in the weight coefficient of the model. During training, it is suggested to damage the most sensitive weights. To determine these weights, test datasets should be passed through the network, and gradients are calculated, which can then be sorted by their absolute values. In the top-k weights with the highest gradient, a single bit was inverted in a random

position. The proposed method generates damage to random weights for testing.

Require: Distribution over disturbances $p(\tau)$; Step size hyperparameters $\alpha$, $\beta$; Number of adaptation steps T.
Pretrain $\Psi$ on original data $D_{base}$
While not done do:
  Select type of disturbance from set {fault injection, evasion adversarial attack, task change}
  Sample disturbance implementations $\tau_i \sim p(\tau)$, $i = \overline{1,n}$
  For i=1,2,…,n do:
    Clone the current parameters: $\dot{\Psi}_{\tau_i}, \dot{W}_{\tau_i} \leftarrow \text{copy}(\Psi, W_{base})$
    If disturbance type is a task change:
      Sample the training and validation data $D_{\tau_i}^{tr}$, $D_{\tau_i}^{val}$ from new task
    else:
      Sample the training and validation data $D_{\tau_i}^{tr}$, $D_{\tau_i}^{val}$ from $D_{base}$
    If disturbance type is a fault injection:
      $\dot{\Psi}_{\tau_i} \leftarrow Fault\_injection(\dot{\Psi}_{\tau_i}, \dot{W}_{\tau_i}, D_{\tau_i}^{tr})$
    If disturbance type is an evasion adversarial attack:
      $D_{\tau_i}^{tr}, D_{\tau_i}^{val} \leftarrow Adversarial\_perturbation(D_{\tau_i}^{tr}, D_{\tau_i}^{val})$
    $\Psi_{\tau_i} \leftarrow SGD_{\Psi}(L_{\tau_i}(\dot{\Psi}_{\tau_i}, \dot{W}_{\tau_i}, D_{\tau_i}^{tr}), T, \alpha)$
    $\Psi \leftarrow \Psi + \beta(\Psi_{\tau_i} - \Psi)$

Figure 5. Pseudocode of meta-learning for object detection network robustness optimization

## 5. Experiments and Results

### 5.1. Experimental Setup

Modern onboard systems of unmanned aerial vehicles (UAVs) are increasingly being equipped with companion computers to enhance their autonomy under challenging conditions. Popular single-board computers include the Radxa Rock 4, Orange Pi 4, and Bana Pi W2, with CPU computational performance ranging from 0.1 to 0.2 TOPS. Currently, dynamic inference can only be executed using CPUs because modern NPUs do not yet support it. It is proposed to train and deploy the neural network using the PyTorch framework because it supports dynamic computational graphs. The input data for the neural network are provided by a camera connected to the single-board computer via the MIPI CSI interface, while communication with the command receiver and flight controller occurs through UART interfaces. It is planned that a slow detector will work in tandem with a fast tracker, such as ByteTrack or a Kalman filter. The network backbone should have a complexity that does not exceed 100 GFLOPs to ensure that predictions can be updated more frequently than once per second. Possible backbones include models such as ViT-S/16 and ViT-B/16.

This study did not explore the impact of different backbone pretraining methods on the efficiency of incorporating dynamic model compression or robustness. Therefore, we selected MAE as one of the most well-researched methods. In this case, pretraining was performed on the ImageNet-1k dataset with a resolution of 512x512 pixels. During fine-tuning, a step-wise learning rate was used, starting at 0.1 and decaying by a factor of 10 after 150 and 250 epochs. VEDAI (Vehicle Detection in Aerial Imagery) is a dataset of annotated images with a resolution of 512x512, used for modeling the main task in supervised learning [44]. DOTA-v2.0, cropped to 512x512, can be used as a data source for auxiliary tasks to model task changes during meta-learning.

The loss function (8) has the following component coefficients: $\lambda_{\text{loc}} = 0.5, \lambda_{\text{cls}} = 0.5, \lambda_{\text{usage}} = 2$. It was proposed to calculate the computational complexity of the model in inference mode as the average FLOPs on the test dataset. The parameter is affected by the parameter $t_k$, which can also be called the dynamic compression rate. Here, parameter is the approximate target rate of execution of each neural network block on a data mini-batch during the training phase.

The mAP (mean Average Precision) for all detection categories was used as an evaluation metric of the object detector on aerial images. The Average Precision of each class was calculated as the area under the Precision-Recall curve [45]. Furthermore, mAP was defined as mAP@0.5, which represents the mean Average Precision when the IoU threshold was set to 0.5.

Considering the elements of randomization, it is proposed to use the average values when evaluating mAP and FLOPs. For this purpose, 100 instances of a specific type of disturbance are generated and applied to the same model or dataset.

### 5.2. Results

Table 1 shows the results of testing the ViT-B/16 model trained on imagenet-1k and aerial images under the influence of Fault Injection (the proportion of modified weights is set by a fault rate of 0.1). Training on its images was performed separately for each Dynamic Compression Rate value to compare the results and select the most compromise option.

Table 2 presents the test results of the ViT-B/16 model pretrained on ImageNet-1k and a set of aerial images on test images distorted by an Adversarial Attack (perturbation level of 3/255 according to the $L_\infty$ norm). Training on aerial images was conducted separately for each Dynamic Compression Rate value to compare the results and select the optimal option. mAP is also used as an evaluation metric for object detectors of aerial images.

An analysis of Tables 1 and 2 shows that reducing the Dynamic Compression Rate, i.e., increasing compression) initially leads to an improvement in the evaluation metric. However, further reduction in the Dynamic Compression Rate results in a loss of the evaluation metric.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Without training or meta-learning under perturbation conditions, reducing the compression rate always decreases the evaluation metrics during testing under the influence of perturbations. However, training or meta-learning under perturbation conditions allows for a slight increase in the evaluation metrics even with reduced compression rates. However, beyond a certain threshold, the evaluation metric begins to decline.

Table 1

Averaged mAP and FLOPs values during testing under fault injection, depending on the pretraining method and the specified Dynamic Compression Rate

| Dyna-mic com-pres-sion rate | Pretrained model under normal condi-tion | | Pretrained model under-fault injection | | Meta-trained model | |
|---|---|---|---|---|---|---|
| | FLOPs ($10^9$) | mAP | FLOPs ($10^9$) | mAP | FLOPs ($10^9$) | mAP |
| 1.0 | 109.2 | 0.69 | 109.2 | 0.72 | 109.2 | 0.75 |
| 0.8 | 94.6 | 0.68 | 92.3 | 0.73 | 91.8 | 0.76 |
| **0.6** | **84.9** | **0.68** | **83.7** | **0.73** | **83.5** | **0.76** |
| 0.4 | 70.4 | 0.67 | 68.9 | 0.72 | 68.0 | 0.74 |
| 0.2 | 60.7 | 0.65 | 58.8 | 0.71 | 57.9 | 0.72 |

Table 2

Averaged mAP and FLOPs values during testing under adversarial attack, depending on the pretraining method and the specified Dynamic Compression Rate

| Dy-namic com-pres-sion rate | Pretrained model under normal condi-tion | | Pretrained model under adversarial attack | | Meta-trained model | |
|---|---|---|---|---|---|---|
| | FLOPs ($10^9$) | mAP | FLOPs ($10^9$) | mAP | FLOPs ($10^9$) | mAP |
| 1.0 | 109.0 | 0.67 | 109.0 | 0.69 | 109.0 | 0.74 |
| 0.8 | 95.1 | 0.66 | 91.8 | 0.70 | 91.1 | 0.75 |
| **0.6** | **85.2** | **0.66** | **82.9** | **0.69** | **84.5** | **0.76** |
| 0.4 | 71.0 | 0.65 | 67.8 | 0.67 | 67.1 | 0.74 |
| 0.2 | 61.2 | 0.63 | 59.1 | 0.65 | 57.2 | 0.72 |

Testing the ViT-B/16 model on the test set of VEDAI data with two different perturbations showed that it achieved an mAP of 0.76 and 84.5 GFLOPs. Thus, reducing FLOPs by 23.5% led to a 1% increase in underweight mAP perturbations compared with the case without compression (Dynamic Compression Rate equal to 1.0). Moreover, the reduction of FLOPs by 22.4% allowed the mAP to be increased in the conditions of perturbed weights by 2% compared to the case without compression (Dynamic compression rate is 1.0).

The analysis given in Table 1 shows that pre-training on modified data under fault injection with the optimal compression rate increased mAP during testing by 7.3% under fault injection. In this case, meta-training based on the results of adaptation to disturbances increased mAP by 11.7% during testing under the influence of fault injection.

The analysis presented in Table 2 shows that pretraining under adversarial attack conditions with an optimal compression rate increased mAP during testing under adversarial attack by 4.5%. In addition, meta-training based on the results of adaptation to disturbances provides a 15.1% increase in mAP during adversarial attack testing.

## 6. Discussion

The results presented in Tables 1 and 2 confirm the reduction of FLOPs of a large network during inference to the level of smaller neural networks due to the dynamic backbone. To implement the dynamic backbone, gate units (2) are added to the network encoders, which add a fixed overhead of 0.015% more floating point operations. With a certain compression rate, the amount of computation in the inference mode decreases, and the mAP under the influence of disturbances does not change significantly or even increase.

In [46], the Yolo8x network on the VEDAI dataset without perturbations provided a mAP of 76.2 and required 104 GFLOPs for inference on one image. In this case, the proposed detector under the influence of fault injection at a compression rate of 0.6 requires only 83.5 GFLOPs but provides a mAP of 0.76. Therefore, the results obtained under disturbance conditions are superior to the known results obtained without disturbance conditions. In addition, if mAP can be tolerated to decrease by 4%, FLOPs can be saved by 32%.

The optimal compression that saves resources and even slightly increases mAP can be explained by the fact that vulnerable and irrelevant parts of the neural network are adaptively switched off. A similar ability of the early exit mechanism to increase fault tolerance and robustness against adversaries was considered in [22, 23].

The analysis presented in Tables 1 and 2 shows that training under perturbation conditions significantly increased mAP in the perturbed inference mode compared to pretraining under normal conditions. Similarly, meta-learning based on adaptation to perturbations allows for an increase in mAP compared to simple pretraining under perturbation conditions. This can be explained by the increased efficiency of gate units. It is likely that with appropriate training, gate units better recognize perturbations that negatively affect the encoder performance.

The primary advantage of methods based on weight quantization, pruning, or knowledge distillation is that the reduced model forms a static computational graph. A static graph can be efficiently deployed on a wide variety of computational devices and frameworks. In contrast, dynamic neural networks are effectively deployed within frameworks such as PyTorch and TensorFlow on CPUs and GPUs. However, currently, commonly used NPUs do not support dynamic computational graphs, which makes it impossible to deploy dynamic inference on such graphs.

## Conclusions

A model of an object detector on aerial images with dynamic inference and optimized robustness is developed for the first time. The proposed model comprises a transformer-based backbone, gate units, and a simplified feature pyramid network with a RetinaNet detection head. Gate units are trained to deactivate transformer encoders that are irrelevant to input data and disturbances. The proposed model reduced FLOPs by more than 22% without loss of the evaluation metric by deactivating some encoders.

For the first time, a training method for an object detector has been developed that combines the RetinaNet loss function with the gate unit loss function and applies meta-learning based on the results of adaptation to various types of synthetic perturbations. The analysis of experimental results showed that meta-training based on adaptation to perturbations increased mAP compared to using either pre-training on annotated data under fault injection or adversarial attack individually.

The proposed model trained using the proposed method under fault injection demonstrated mAP at the Yolo8x level without fault injection. In addition, the proposed model required 24% fewer FLOPs than Yolo8x, indicating higher computational efficiency than other popular approaches.

**Contribution of authors:** development of conceptual provisions and methodology of research, formulation of conclusions – **Viacheslav Moskalenko**; Review and analysis of references; development of software for modeling – **Artem Korobov**; development of mathematical models, analysis of research results – **Yuriy Moskalenko**.

**Conflict of interest:** the authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

**Financing:** This study was conducted without financial support.

**Data availability:** The manuscript has associated data in a data repository: The VEDAI and DOTA-v2.0 datasets are available in [38] and can be found here https://downloads.greyc.fr/vedai/, https://captain-whu.github.io/DOTA/dataset.html, accessed on September 17, 2022. The ILSVRC2012 dataset can be found at http://image-net.org/challenges/LSVRC/2012/index, accessed on 17 September 2022.

**Use of artificial intelligence:** the authors used artificial intelligence technologies to provide verified results. The writing of the article text was performed without the use of artificial intelligence technologies.

**Acknowledgments:** the authors thank the Ministry of Education and Science of Ukraine for providing support to the Laboratory of Intellectual Systems within the framework of research project No. 0124U000548 “Information Technology for Ensuring the Resilience of the Intelligent Onboard System of Small-Sized Aircraft” (2024–2026).

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

## References

1. Polina, A. M., Suparwito, H., & Kumalasanti, R. A. Aerial object detection analysis: Challenges and preliminary results. *E3S Web of Conferences*, 2024, vol. 475, article no. 02017. DOI: 10.1051/e3sconf/202447502017.

2. Marinó, G. C., Petrini, A., Malchiodi, D., & Frasca, M. Deep neural networks compression: A comparative survey and choice recommendations. *Neurocomputing*, 2023, vol. 520, pp. 152–170. DOI: 10.1016/j.neucom.2022.11.072.

3. Syed, R., Ulbricht, M., Piotrowski, K., & Krstic, M. A Survey on Fault-Tolerant Methodologies for Deep Neural Networks. *Pomiary Automatyka Robotyka*, 2023, vol. 27, iss. 2, pp. 89–98. DOI: 10.14313/par_248/89.

4. Chen, X., Huang, W., Peng, Z., Guo, W., & Zhang, F. Diversity supporting robustness: Enhancing adversarial robustness via differentiated ensemble predictions. *Computers & Security*, 2024, vol. 142, article no. 103861. DOI: 10.1016/j.cose.2024.103861.

5. Niu, Z., Chen, Z., Li, L., Yang, Y., Li, B., & Yi, J. On the Limitations of Denoising Strategies as Adversarial Defenses. *arXiv*, 2020. DOI: 10.48550/ARXIV.2012.09384.

6. Lust, J., & Condurache, A. P. Efficient detection of adversarial, out-of-distribution and other misclassified samples. *Neurocomputing*, 2022, vol. 470, pp. 335–343. DOI: 10.1016/j.neucom.2021.05.102.

7. . Lu, Z., Sun, H., Ji, K., & Kuang, G. Adversarial Robust Aerial Image Recognition Based on Reactive-Proactive Defense Framework with Deep Ensembles. *Remote Sensing*, 2023, vol. 15, iss. 19, article no. 4660. DOI: 10.3390/rs15194660.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

8. Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. A survey of modern deep learning based object detection models. *Digital Signal Processing*, 2022, vol. 126, article no. 103514. DOI: 10.1016/j.dsp.2022.103514.

9. Han, Y., Huang, G., Song, S., Yang, L., Wang, H., & Wang, Y. Dynamic Neural Networks: A Survey. *arXiv*, 2021. DOI: 10.48550/ARXIV.2102.04906.

10. Meng, L., Li, H., Chen, B.-C., Lan, S., Wu, Z., Jiang, Y.-G., & Lim, S.-N. AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, article no. 01199. DOI: 10.1109/cvpr52688.2022.01199.

11. Tang, G., Ni, J., Zhao, Y., Gu, Y., & Cao, W. A Survey of Object Detection for UAVs Based on Deep Learning. *Remote Sensing*, 2023, vol. 16, iss. 1, article no. 149. DOI: 10.3390/rs16010149.

12. Paul, S., & Chen, P.-Y. Vision Transformers Are Robust Learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, iss. 2, pp. 2071–2081. DOI: 10.1609/aaai.v36i2.20103..

13. Li, Y., Fan, Q., Huang, H., Han, Z., & Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones*, 2023, vol. 7, iss. 5, article no. 304. DOI: 10.3390/drones7050304.

14. Wu, W., Liu, A., Hu, J., Mo, Y., Xiang, S., Duan, P., & Liang, Q. EUAVDet: An Efficient and Lightweight Object Detector for UAV Aerial Images with an Edge-Based Computing Platform. *Drones*, 2024, vol. 8, iss. 6, article no. 261. DOI: 10.3390/drones8060261.

15. Kim, J., Chang, S., & Kwak, N. PQK: Model Compression via Pruning, Quantization, and Knowledge Distillation. *arXiv*, 2021. DOI: 10.48550/ARXIV.2106.14681.

16. Hawks, B., Duarte, J., Fraser, N. J., Pappalardo, A., Tran, N., & Umuroglu, Y. Ps and Qs: Quantization-Aware Pruning for Efficient Low Latency Neural Network Inference. *Frontiers in Artificial Intelligence*, 2021, vol. 4. DOI: 10.3389/frai.2021.676564.

17. Haque, M., & Yang, W. Dynamic Neural Network is All You Need: Understanding the Robustness of Dynamic Mechanisms in Neural Networks. *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2023, pp. 1489–1498. DOI: 10.1109/iccvw60793.2023.00163.

18. Moskalenko, V., Kharchenko, V., Moskalenko, A., & Kuzikov, B. Resilience and Resilient Systems of Artificial Intelligence: Taxonomy, Models and Methods. *Algorithms*, 2023, vol. 16, iss. 3, article no. 165. DOI: 10.3390/a16030165.

19. Moskalenko, V., & Moskalenko, A. Neural network based image classifier resilient to destructive perturbation influences – architecture and training method. *Radioelectronic and Computer Systems*, 2022, iss. 3, pp. 95–109. DOI: 10.32620/reks.2022.3.07.

20. Lu, Z., Sun, H., Ji, K., & Kuang, G. Adversarial Robust Aerial Image Recognition Based on Reactive-Proactive Defense Framework with Deep Ensembles. *Remote Sensing*, 2023, vol. 15, iss. 19, article no. 4660. DOI: 10.3390/rs15194660.

21. Fang, C., Song, Y., Guan, F., Liang, F., & Yang, L. A Robust Complex-Valued Deep Neural Network for Target Recognition of UAV SAR Imagery. *IEEE Journal on Miniaturization for Air and Space Systems*, 2023, vol. 4, iss. 2, pp. 175–185. DOI: 10.1109/jmass.2023.3247586.

22. Wang, J., Zhang, Z., Wang, M., Qiu, H., Zhang, T., Li, Q., Li, Z., Wei, T., & Zhang, C. Aegis: Mitigating Targeted Bit-flip Attacks against Deep Neural Networks. *arXiv*, 2023. DOI: 10.48550/ARXIV.2302.13520.

23. Haque, M., & Yang, W. Dynamic Neural Network is All You Need: Understanding the Robustness of Dynamic Mechanisms in Neural Networks. *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2023, pp. 1489-1498. DOI: 10.1109/iccvw60793.2023.00163.

24. Cavagnero, N., Santos, F. D., Ciccone, M., Averta, G., Tommasi, T., & Rech, P. Transient-Fault-Aware Design and Training to Enhance DNNs Reliability with Zero-Overhead. *2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2022. DOI: 10.1109/iolts56730.2022.9897813.

25. Dong, J., Qiu, H., Li, Y., Zhang, T., Li, Y., Lai, Z., Zhang, C., & Xia, S.-T. One-bit Flip is All You Need: When Bit-flip Attack Meets Model Training. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 4665-4675. DOI: 10.1109/iccv51070.2023.00432.

26. Cao, H., & Xue, M. Adversarial Training for Better Robustness. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer Nature Switzerland*, 2023, pp. 75-84. DOI: 10.1007/978-3-031-35982-8_6.

27. Bai, T., Luo, J., Zhao, J., Wen, B., & Wang, Q. Recent Advances in Adversarial Training for Adversarial Robustness. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, 2021, pp. 4312–4321. DOI: 10.24963/ijcai.2021/591.

28. Athalye, A., Carlini, N., & Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *arXiv*, 2018. DOI: 10.48550/ARXIV.1802.00420.

29. Qiu, P., Wang, Q., Wang, D., Lyu, Y., Lu, Z., & Qu, G. Mitigating Adversarial Attacks for Deep Neural Networks by Input Deformation and Augmentation. *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, pp. 157–162. DOI: 10.1109/aspdac47756.2020.9045107.

30. Zhang, B., Tondi, B., Lv, X., & Barni, M. Challenging the Adversarial Robustness of DNNs Based on Error-Correcting Output Codes. *Security and Communication Networks*, 2020, vol. 2020, pp. 1–11. DOI: 10.1155/2020/8882494.

31. Wang, R., Xu, K., Liu, S., Chen, P.-Y., Weng, T.-W., Gan, C., & Wang, M. On Fast Adversarial Robustness Adaptation in Model-Agnostic Meta-Learning (Version 1). *arXiv*, 2021. DOI: 10.48550/ARXIV.2102.10454.

32. Maurício, J., Domingues, I., & Bernardino, J. Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: *A Literature Review. Applied Sciences*, 2023, vol. 13, iss. 9, article no. 5521. DOI: 10.3390/app13095521.

33. Wang, L., & Tien, A. Aerial Image Object Detection with Vision Transformer Detector (ViTDet). *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, 2023. DOI: 10.1109/igarss52108.2023.10282836.

34. Li, Y., Mao, H., Girshick, R., & He, K. Exploring Plain Vision Transformer Backbones for Object Detection. *Lecture Notes in Computer Science, Springer Nature Switzerland*, 2022, pp. 280–296. DOI: 10.1007/978-3-031-20077-9_17.

35. Yuldashev, Y., Mukhiddinov, M., Abdusalomov, A. B., Nasimov, R., & Cho, J. Parking Lot Occupancy Detection with Improved MobileNetV3. *Sensors*, 2023, vol. 23, iss. 17, article no. 7642. DOI: 10.3390/s23177642.

36. Nawaz, S. A., Li, J., Bhatti, U. A., Shoukat, M. U., & Ahmad, R. M. AI-based object detection latest trends in remote sensing, multimedia and agriculture applications. *Frontiers in Plant Science*, 2022, vol. 13. DOI: 10.3389/fpls.2022.1041514.

37. Assran, M., Caron, M., Misra, I., Bojanowski, P., Bordes, F., Vincent, P., Joulin, A., Rabbat, M., & Ballas, N. Masked Siamese Networks for Label-Efficient Learning. *Lecture Notes in Computer Science, Springer Nature Switzerland*, 2022, pp. 456–473. DOI: 10.1007/978-3-031-19821-2_26.

38. Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., & Joulin, A. Emerging Properties in Self-Supervised Vision Transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, article no. 00951. DOI: 10.1109/iccv48922.2021.00951.

39. Pinetsuksai, N., Kittichai, V., Jomtarak, R., Jaksukam, K., Tongloy, T., Boonsang, S., & Chuwongin, S. Development of Self-Supervised Learning with Dinov2-Distilled Models for Parasite Classification in Screening. *2023 15th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2023, pp. 323-328. DOI: 10.1109/icitee59582.2023.10317719.

40. Moskalenko, V. V. Model-Agnostic Meta-Learning for Resilience Optimization of Artificial Intelligence System. *Radio Electronics, Computer Science, Control*, 2023, iss. 2, article no. 79. DOI: 10.15588/1607-3274-2023-2-9.

41. Kulkarni, U., Meena, S. M., Hallyal, R., Sulibhavi, P., Sunil, V. G., Guggari, S., & Shanbhag, A. R. Optimisation of deep neural network model using Reptile meta learning approach. *Cognitive Computation and Systems, Institution of Engineering and Technology (IET)*, 2023. DOI: 10.1049/ccs2.12096.

42. Kotyan, S., & Vargas, D. V. Adversarial robustness assessment: Why in evaluation both L0 and L∞ attacks are necessary. *PLOS ONE*, 2022, vol. 17, iss. 4, article no. e0265723. DOI: 10.1371/journal.pone.0265723.

43. Li, G., Pattabiraman, K., & DeBardeleben, N. TensorFI: A Configurable Fault Injector for TensorFlow Applications. *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2018. DOI: 10.1109/issrew.2018.00024.

44. Zhao, Y., Sun, H., & Wang, S. Small Object Detection in Medium–Low-Resolution Remote Sensing Images Based on Degradation Reconstruction. *Remote Sensing*, 2024, vol. 16, iss. 14, article no. 2645. DOI: 10.3390/rs16142645.

45. Padilla, R., Passos, W. L., Dias, T. L. B., Netto, S. L., & da Silva, E. A. B. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics*, 2021, vol. 10, iss. 3, article no. 279. DOI: 10.3390/electronics10030279.

46. Shen, L., Lang, B., Song, Z. Infrared Object Detection Method Based on DBD-YOLOv8. *IEEE Access*, 2023, vol. 11, pp. 145853–145868. DOI: 10.1109/access.2023.3345889.

*Received 12.07.2024, Accepted 20.08.2024*

## ДЕТЕКТУВАННЯ ОБ'ЄКТІВ З РАЦІОНАЛЬНОЮ РОБАСТНІСТЮ ДЛЯ АЕРОЗОБРАЖЕНЬ: МОДЕЛЬ ТА МЕТОД ЗАБЕЗПЕЧЕННЯ

***В. В. Москаленко, А. Г. Коробов, Ю.В. Москаленко***

Нейромережеві детектори об'єктів набувають широкого застосування для аналізу аеровідеоспостережень. Зростає кількість практично корисних прикладів використання, що потребують оброблення даних на борту безпілотного літального апарату чи іншої обмеженої в ресурсах платформи. Проте вразливість нейронних мереж до протиборчого шуму, новизни в даних та ін'єкції помилок у вагові коефіцієнти обмежує функціональні можливості і надійність подібних рішень. Розроблення моделей детекторів і методів їх навчання, що одночасно забезпечать обчислювальну ефективність та робастності до збурень є актуальною науковою задачею. **Предметом** дослідження у статті є модель і метод забезпечення робастності нейромережевих систем детектування об'єктів на аеровідеоспостереженнях в умовах обмежених ресурсів. **Метою** дослідження є розроблення моделі і методу забезпечення раціональної з точки зору обчислювальних ресурсів робастності детекторів об'єктів на аерозображеннях. Використовуваними **методами** є: методи динамічних нейронних мереж, методи оптимізації робастності і резільєнтності нейромереж. Отримано такі **результати.** Розроблена модель детектора з трансформаторним екстрактором ознак ViT-B/16, що модифікований вентельними модулями для реалізації динамічного інференсу. Модель була навчена на наборі даних VEDAI і мета-навчена на результатах

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

адаптації до різнотипних збурень. Було протестовано модель з різним способом навчання на стійкість до випадкової інверсії бітів у вагах, де пропорція модифікованих ваг становить 10%. Крім того, було протестовано модель з різним способом навчання на стійкість до протиборчих атак чорного ящика з амплітудою збурення до 3/255 відповідно до L∞ норми. **Висновки.** Вперше розроблено модель детектора об'єктів на аерозображеннях з динамічним інференсом та оптимізованою робастністю. Ця модель основана на використанні вентильних модулів, а також спрощеної мережі піраміди ознак з RetinaNet головкою детектування. Вентильні модулі навчені деактивувати нерелевантні для вхідних даних і збурень енкодери трансформера. Запропонована модель без втрати точності дозволяла зменшити кількість операцій з плаваючою крапкою більше ніж на 22% за рахунок деактивації частини енкодерів. Вперше розроблено метод навчання детектора, що полягає у поєднанні функції втрат RetinaNet з функцією втрат вентильних модулів та у застосуванні мета-навчання на результатах адаптації до різнотипних синтетичних збурень. Аналіз результатів експериментів показав, що запропонований підхід дозволяє під час тестування в умовах впливу ін'єкції помилок у ваги мережі підвищити усереднену точність на 11,7%, а під час тестування в умовах впливу протиборчих атак підвищення усередненої точності на 15,1%.

**Ключові слова**: детектування об'єктів; робастність; протиборчі атаки; ін'єкції несправностей; мета-навчання.

**Москаленко В'ячеслав Васильович** – канд. техн. наук, доц., доц. каф. комп'ютерних наук, Сумський державний університет, Суми, Україна.

**Коробов Артем Геннадійович** – канд. техн. наук, старший викладач каф. комп'ютерних наук, Сумський державний університет, Суми, Україна.

**Москаленко Юрій Васильович** – асп. каф. комп'ютерних наук, Сумський державний університет, Суми, Україна.

**Viacheslav Moskalenko** – PhD, Associate Professor at the Computer Science Department of Sumy State University, Sumy, Ukraine,
e-mail: v.moskalenko@cs.sumdu.edu.ua, ORCID: 0000-0001-6275-9803, Scopus Author ID: 57189099775.

**Artem Korobov –** PhD, Senior Lecturer at the Computer Sciences Department of Sumy State University, Sumy, Ukraine,
e-mail: a.korobov@cs.sumdu.edu.ua, ORCID: 0000-0003-3239-1977, Scopus Author ID: 57196299250.

**Yuriy Moskalenko** – PhD Student of the Computer Science Department of Sumy State University, Sumy, Ukraine,
e-mail: yuriy.mosk@gmail.com, ORCID: 0009-0002-3635-3337.