UDC 004.891.032.26:629.7.01.066

RESOURCE-EFFICIENT, ROBUST, AND ADAPTIVE OBJECT DETECTION IN UAV IMAGERY

Moskalenko V. V. – PhD, Associate Professor, Associate Professor of Computer Science department, Sumy State University, Sumy, Ukraine. ROR: https://ror.org/01w60n236. ORCID: https://orcid.org/0000-0001-6275-9803.

Moskalenko A. S. – PhD, Associate Professor, Associate Professor of Computer Science department, Sumy State University, Sumy, Ukraine. ROR: https://ror.org/01w60n236. ORCID: https://orcid.org/0000-0003-3443-3990.

Moskalenko Y. V. – Post-graduate student, Sumy State University, Sumy, Ukraine. ROR: https://ror.org/01w60n236. ORCID: https://orcid.org/0000-0002-9121-7832.

Vatsenko A. V. – Post-graduate student, Sumy State University, Sumy, Ukraine. ROR: https://ror.org/01w60n236. ORCID: https://orcid.org/0009-0007-3278-0194.

ABSTRACT

Context. Ensuring robust, adaptable, and compute-efficient object detection in UAV aerial imagery under distribution shifts, structured and unstructured noise, and strict onboard latency/energy budgets is an urgent scientific task.

A compute-aware detector and a complementary training/adaptation method that integrate a dynamic transformer backbone with gate units, parameter-efficient adapters, and resource-bounded test-time adaptation to sustain accuracy under realistic perturbations and domain shift.

Objective. Development of a model and method for object detection in aerial imagery that jointly provide robustness and adaptability while meeting embedded compute and real-time constraints typical of onboard UAV systems.

Methods. The approach combines dynamic neural networks with Gumbel-Softmax gate units over a ViT-T/16 backbone, a Simple FPN and a RetinaNet-like one-stage head, budget-aware losses that target a desired dynamic compression rate, structured procedural noise (Perlin, Gabor, Worley) for robustness training, LeakyReLU6 with a straight-through estimator for stable gradients, and test-time adaptation via objectness-weighted marginal-entropy minimization on lightweight adapters.

Results. On VEDAI, a gated ViT-T/16 detector reaches mAP@0.5 of 0.77 at ~5.0 GFLOPs and 17.8 FPS, rising to 0.79 with adapters and TTA, whereas a static counterpart attains 0.74 at 9.6 GFLOPs and 10.6 FPS; pretraining with procedural noise lifts accuracy further to 0.80 (gated) and 0.82 (gated+TTA) with minimal compute overhead. Under domain shift (trained on VisDrone, evaluated on VEDAI), dynamic gating and TTA improve mAP from 0.54 to 0.60 without noise pretraining and up to 0.66 with it, sustaining ~5.4–5.6 GFLOPs and ~16–17 FPS within an 8–10 GFLOPs budget on 4×A76 CPUs.

Conclusions. The proposed object detection model and method – combining dynamic gating, perturbation-aware training, and budgeted test-time adaptation – reduce average compute while increasing robustness and adaptability, yielding a superior accuracy-throughput trade-off for UAV onboard deployment under real-world disturbances and distribution shifts.

KEYWORDS: object detection, robustness, adaptability, adversarial procedural noise, dynamic neural network.

ABBREVIATIONS

ASIC is an application-specific integrated circuit;
CNN is a convolutional neural network;
DINO is a self-distillation with no labels;
FLOP is a floating-point operations per second;
FPN is a feature pyramid network;
FPS are frames per second;
GFLOP is a billion FLOPS;
IoU is an intersection over union metric;
LN is a layernorm;
MAE is a masked auto-encoding;
mAP is a mean average precision
MLP is a multi-layer perceptron;
MSA is a multi-head self-attention;
MSN is a masked siamese networks;
NPU is a neural processing unit
SoC is a system on a chip;
STE is a straight-through estimator;
TTA is a test-time adaptation;
UAV is an unmanned aerial vehicles;
VEDAI is a vehicle detection in aerial imagery.

NOMENCLATURE

a_k is an adapter function;

B_{FLOPs} is a maximum average number of floating-point operations per frame;

B_{FPS} is a minimum required throughput in frames per second;

C is a number of classes;

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);

f_θ is a detector models with a transformer backbone, gate units for dynamic computation, parameter-efficient adapters, and a test-time adaptation mechanism;

F is a set of random control points (feature points);

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

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

$g(x, y)$ is a Gabor noise function;

$G_{per}(x, y)$ is a sinusoidal transformation for Perlin noise;

DOI 10.15588/1607-3274-2026-2-8

k is a k-th block depending on the conditions;

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

L is a composite loss function;

L_{usage} is a set of parameters of tuners;

L_{loc} is a bounding box regression loss;

L_{cls} is a classification loss;

L_{usage} is a function that calculates the deviation of the desired dynamic compression rate of the neural network from the real one;

L_{TTA} is a loss function for Test-Time Adaptation takes the following form;

n is a size of the training mini-batch;

$o_a^{(b)}$ is an objectness logit at anchor a under augmentation b;

p_t is a probability of predicting the i-th class;

$p(\cdot)$ is a classic 2D Perlin noise function;

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

$p_a^{(b)}$ is a class-probability vector for anchor under augmentation b obtained by applying a softmax to the class logits $z_a^{(b)}$;

$S_{per}(x,y)$ is a perlin noise;

$S_{gab}(x,y)$ is a resulting Gabor noise;

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

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

w_a is an objectness weight (downweights clear background);

$W(x,y)$ is a Worley noise (cellular noise);

(x_i, y_i) are coordinates of randomly selected kernel placement points;

(x,y) is a pixel or point coordinates in space;

z_{k-1} is an input tensor;

z_k is an output tensor;

γ is a shaping exponent that controls how strongly the average objectness influences the weight (lower γ softens background down-weighting), $\gamma \in [0,1]$;

γ is a focus parameter;

$\in$ is a noise amplitude parameter;

θ^* is a detector parameters;

λ is a wavelength (harmonic period);

λ_{loc} is a trade-off coefficients between different components of the composite loss function;

λ_{cls} is a trade-off coefficients between different components of the composite loss function;

λ_{usage} is a trade-off coefficients between different components of the composite loss function;

λ_x is a wavelength parameters along x axis;

λ_y is a wavelength parameters along y axe;

ξ is a number of orientations (degree of isotropy);

$\sigma(\cdot)$ is a logistic sigmoid function;

σ is a Gaussian envelope width;

ϕ_{sine} is a sinusoidal frequency parameter;

ω is an orientation;

Ω is a number of octaves (i.e., detail levels).

INTRODUCTION

Aerial imagery for UAV-based perception exhibits extreme diversity in viewing geometry, object scale, and background clutter, together with frequent distribution shifts driven by altitude, platform dynamics, sensor changes, illumination, weather, and seasonal patterns [1]. These factors strain both data requirements and compute budgets on-board, where energy, memory, and latency are strictly constrained. While compression techniques (pruning, quantization, distillation) reduce model size and operations, they often degrade resilience to perturbations and are brittle under domain shift, particularly for small or densely packed targets typical in aerial scenes [2].

Robustness in vision models has been pursued through ensembling, denoising, adversarial training, redundancy of pathways, and training on augmented or corrupted data [3, 4]. However, many of these strategies assume ample resources or cloud inference, and their benefits in the onboard UAV regime are limited by added compute and memory overheads. Moreover, robustness evaluations commonly emphasize image classification benchmarks, leaving object detection where localization errors under noise or shift can be safety-critical – comparatively underexplored.

Recent progress in object detection has focused on architectural refinements (convolutional backbones, transformer-based detectors) and micro-architectural changes that improve multi-scale localization and classification accuracy [5]. In parallel, dynamic inference and conditional computation have emerged to better allocate computation across inputs or spatial regions, reporting promising efficiency gains [6]. Yet much of this evidence comes from classification tasks and desktop settings; systematic studies on detection-specific robustness (e.g., stability of proposals, localization under corruptions, routing consistency under nuisance variation) remain scarce. Likewise, parameter-efficient transfer methods (e.g., adapters, LoRA) demonstrate strong data and compute-efficiency in domain adaptation, but their interaction with dynamic execution and their behavior under tight real-time constraints typical of UAVs are insufficiently characterized.

Another active direction is TTA, which aims to cope with distribution shift without labels by optimizing unsupervised objectives during deployment [7]. Although TTA can recover accuracy under moderate shift, its

DOI 10.15588/1607-3274-2026-2-8

safety, stability, and compute footprint on embedded platforms remain open issues particularly for detectors, where adaptation must avoid catastrophic calibration drift and preserve real-time throughput. Existing evaluations seldom combine TTA with stringent FLOPs/latency budgets or with stressors resembling real aerial artifacts such as atmospheric effects, compression noise, or structured procedural textures.

There is a research gap in simultaneously ensuring robustness, adaptability, and reducing the computational complexity of object detection models in aerial imagery. Prior work typically optimizes one or two of these axes in isolation – e.g., efficiency without adaptation, adaptation without robustness guarantees, or robustness with unconstrained resources – leaving UAV systems vulnerable to realistic shifts, structured and unstructured noise, and onboard compute limits.

The aim of this research is to establish a compute-aware framework for robustness and adaptability in object detectors for aerial imagery, suitable for onboard UAV deployment under strict latency and energy constraints.

Robustness is the ability of a detector to maintain functionality under destructive perturbations (noise, environmental shifts). We quantify it via residual performance under corruption and report it jointly with compute and latency. Adaptability is the ability of a detector to recover or preserve accuracy under distribution shift (changes in altitude, optics, weather, background, platform) without full retraining, through (I) conditional use of capacity at inference time and/or (II) lightweight parameter updates that respect onboard budgets.

The key issues are as follows:

– analysis of existing solutions for ensuring robustness, adaptability, and reduced computational complexity in object detection for aerial imagery, with emphasis on onboard UAV constraints;

– development of a dynamic (adaptive) neural network detector for aerial imagery that supports conditional computation and parameter-efficient adaptation, while operating under high observation variability and resisting diverse noise types and network weight corruption;

– development of a training and online-evaluation methodology to ensure robustness and adaptability, including adversarial training, corruption curricula with procedural noises (Worley, Perlin, Gabor), and budget-aware objectives; additionally, support test-time adaptation via unsupervised objectives (e.g., entropy minimization) with stability safeguards.

Structurally, the work consists of the following sections. The related works are analyzed in the Section 2. The Section 3 presents a new computational efficient model for object detection on aerial images. The Section 4 describes a new training and tuning method used to provide computational efficiency, model robustness and adaptability to input perturbations. The Section 5 describes the experimental results of testing of 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 the directions of future research.

1 PROBLEM STATEMENT

Suppose we are given: a labelled dataset $D = \{(x_i, y_i)\}$ of UAV aerial images x_i with ground-truth bounding boxes and classes y_i; a set τ of corruption and shift operators $\tau : x \mapsto \tau(x)$ that model realistic perturbations of UAV imagery (changes in altitude, optics, illumination, weather, background, as well as structured noises such as Perlin, Gabor, and Worley fields); a target hardware platform with an onboard compute – latency budget $B = (B_{FLOPs}, B_{FPS})$, where B_{FLOPs} is the maximum average number of floating-point operations per frame and B_{FPS} is the minimum required throughput in frames per second; a family of detector models f_θ with a transformer backbone, gate units for dynamic computation, parameter-efficient adapters, and a TTA mechanism.

The problem of resource-efficient, robust, and adaptive object detection in UAV imagery consists in finding a detector f_{θ^*} and an associated adaptation policy such that maximize the detection quality on UAV data, measured by the mean Average Precision at IoU 0.5 (mAP@0.5). The optimization criterion is to maximize the residual detection quality under perturbations and shift, for example:

$$mAP_{rob}(f_\theta) \to \max_\theta . \quad (1)$$

The constraints are:

$$C(f_\theta) \le B_{FLOPs}, F(f_\theta) \ge B_{FPS} . \quad (2)$$

The output variables of the problem are

– the detector parameters θ^* (backbone, FPN, detection head, gate units, adapters);

– the dynamic gating policy and TTA parameters that define which blocks are executed and how adapters are updated at test time.

2 REVIEW OF THE LITERATURE

Two-stage (e.g., Faster R-CNN) and one-stage (e.g., YOLO, RetinaNet) detectors based on CNNs have long been the mainstay of object detection [7, 8]. These models achieve high accuracy across scales via multi-level feature pyramids and specialized heads for classification and localization. In UAV aerial imagery, however, the high density of small targets, extreme multi-scale variation, and complex backgrounds increase architectural complexity and raise demands on computation and memory, which conflicts with onboard real-time requirements.

DOI 10.15588/1607-3274-2026-2-8

Vision Transformers (ViTs) and their derivatives improve integration of local and global context and often generalize well on large datasets [9]. Nevertheless, transformers are typically less compute-efficient than CNNs, which challenges real-time deployment on UAV hardware. Furthermore, much of the literature assumes stationary datasets and offline inference; adaptation at deployment time (e.g., TTA) to cope with altitude, illumination, or sensor shifts is underexplored for detectors, despite being highly relevant for UAV missions operating in non-stationary environments [10].

Mainstream CNN- and ViT-based detectors deliver accuracy, but their resource footprint and limited attention to deployment-time adaptation hinder practicality for onboard UAV perception.

Architectures such as Bi-PAN-FPN [11] and EUAVDet [12] improve the speed-accuracy trade-off for aerial images by optimizing multi-scale fusion (e.g., lightweight necks and ghost modules), which helps detect small objects at reduced FLOPs. In parallel, model compression – quantization, pruning, and knowledge distillation – remains a standard path to fit detectors into constrained platforms [13].

However, aggressive compression can reduce robustness to distribution shifts and corruptions. This has motivated dynamic inference approaches – early-exit networks and gate-based conditional computation – that allocate compute adaptively across inputs or spatial regions [14, 15]. While such methods lower average cost, their interaction with TTA and their behavior under strict onboard latency/energy budgets remain insufficiently characterized for detection tasks.

Beyond static efficiency, a growing body of work investigates parameter-efficient transfer and test-time/online adaptation to handle distribution shift without full retraining. Yet most evaluations target classification and desktop-class hardware; for UAV detectors, open issues include stability of adaptation, budgeted update policies, and compatibility with conditional compute under real-time constraints.

Efficient neck/FPN designs and compression reduce cost, and dynamic inference offers adaptive compute; still, resource-bounded TTA for detectors is not comprehensively studied, especially under realistic UAV constraints.

Despite architectural progress, neural networks remain vulnerable to adversarial attacks, sensor faults, and out-of-distribution inputs – risks amplified in UAV operations. Both CNNs and ViTs have been shown to be sensitive to such disturbances [16]. Reported countermeasures span ensemble defenses [17], sensor-specific designs (e.g., SAR-oriented approaches) [18], dynamic inference with incidental robustness benefits [19], adversarial training and corruption-focused training protocols [20], as well as input preprocessing, postprocessing, and architectural modifications [21]. Meta-learning has also been explored to improve classifier resilience [22]. Much of this evidence, however, originates from classification settings and desktop-class hardware, with limited emphasis on real-time constraints and detection-specific failure modes relevant to onboard UAV platforms.

Dynamic neural networks have been investigated as a means to couple adaptability with efficiency. Through conditional computation – via gate units or early exits – models can select subpaths conditioned on the input, effectively yielding dynamic compression by skipping non-essential blocks while allocating capacity to harder regions or instances [17, 22, 23]. When routing policies are trained or regularized under corruptions and domain shifts, studies report a tendency to prefer more stable, noise-tolerant blocks, suggesting that conditional execution can contribute to robustness while reducing average compute. Such input- and region-dependent allocation of depth and receptive fields is relevant for aerial scenes with dense small objects and rapidly varying observation conditions.

TTA has emerged as a popular strategy to handle distribution shift without labels at deployment. Prior work examines unsupervised objectives and lightweight parameter updates to improve performance under changing conditions (e.g., altitude, optics, illumination, background), typically constraining the set of updated parameters and the update frequency to respect latency and energy budgets [7, 24]. In detection, the literature increasingly considers how TTA interacts with conditional computation and streaming data, though comprehensive evaluations under strict onboard constraints remain relatively sparse.

Robustness studies for aerial sensing also emphasize exposure to diverse, realistic perturbations. Beyond classical corruptions, researchers increasingly use synthetic structured noise fields – such as Worley, Gabor, and Perlin to approximate real structured disturbances (e.g., texture variations, atmospheric effects, codec artifacts, sensor patterns) [25]. These synthetic fields are employed within adversarial training or data augmentation to emulate in-the-wild variability and to probe localization stability, proposal consistency, and small-object recall.

Taken together, the literature points to complementary lines of progress – dynamic inference for adaptive compute, TTA for deployment-time tuning/fine-tuning under shift, and training/evaluation protocols that incorporate synthetic structured noise to mimic real conditions. At the same time, detector-focused, resource-bounded studies that jointly assess these elements under real-time power, memory, and stability constraints typical of onboard UAV systems remain limited.

3 MATERIALS AND METHODS

As the backbone, we use a Transformer – ViT-T/16 or ViT-S/16 – pre-trained on a large dataset [26]. To improve computational efficiency and adaptability to context and disturbances, we introduce gating units that dynamically disable irrelevant or misleading Transformer encoders and their corresponding parallel adapters. In other words, only the relevant features are computed by

DOI 10.15588/1607-3274-2026-2-8

an activated subset of layers. The proposed dynamic backbone architecture is shown in Fig. 1. The backbone is composed of sequential blocks (structural units). Each network block includes a MSA block, a MLP block with skip connections, a gating unit g_k that activates or deactivates the k-th block depending on the conditions, and an adapter unit a_k for test-time adaptation to disturbances.

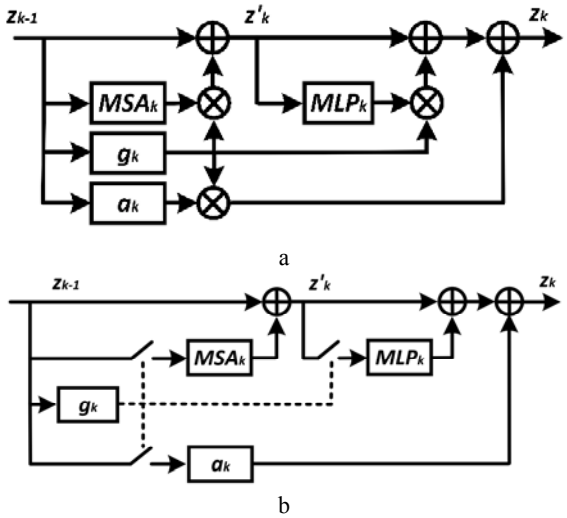

Figure 1 – Schematic illustration of the structural unit of a dynamic visual transformer based backbone:
a – training mode; b – inference mode

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:

$$z_k^{'} = z_{k-1} + g_k(z_{k-1})MSA_k(z_{k-1}), \quad (3)$$

$$z_k = z_k^{'} + g_k(z_{k-1})MLP_k(z_k^{'}) + g_k(z_{k-1})a_k(z_{k-1}). \quad (4)$$

The computational graph for inference can be defined as follows:

$$z_k = \begin{cases} z_{k-1}, if g_k(z_{k-1}) = 0, \\ \tilde{z}_k + MLP_k(\tilde{z}_k) + a_k(z_{k-1}), \\ if g_k(z_{k-1}) = 1, \end{cases} \quad (5)$$

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

Based on the functional purpose of the gate unit and the specifics of training multilayer neural networks, it should have the following properties [27]:

– low computational complexity compared to the 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 structure of the gate unit in the training and inference modes. The addition of Gumbel noise $G = -\log(-\log(U))$, where $U \sim \text{Unif}[0, 1]$, 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.

In the experiments, the gate unit model is 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 [28].

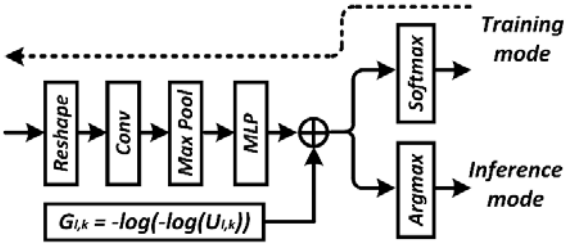

Figure 2 – The architecture of the gate unit

The larger the model, the more computationally complex it is to fine-tune for adaptation to new conditions. Moreover, there is a potential risk of catastrophic forgetting under the influence of new information. Therefore, it is proposed to attach a_k tuners to the model, which can be computationally efficient in fine-tuning [29]. In this case, the weight coefficients of the model remain frozen. The parallel method of connecting a tuner (adapter) to the frozen blocks of the model is the most convenient and versatile approach. Various tuner (adapter) architectures have become popular in the literature, with the most computationally efficient ones shown in Fig. 3.

DOI 10.15588/1607-3274-2026-2-8

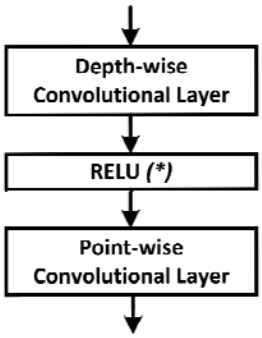

Figure 3 – Parameter-efficient adapter architecture for each block of backbone

Efficient adaptation of the task-agnostic plain backbone to a specific task of detecting objects in aerial images requires adding a specific task-specific bottleneck to the backbone output. This bottleneck should separate the features that are most convenient for encoding information about detected objects of different sizes. In [30], the so-called Simple Feature Piramid Network was proposed, which forms 4 different scale feature maps (Fig. 4). The 1/32 scale is built by stride-2 2×2 max pooling (average pooling or convolution works similarly). The 1/16 scale simply uses the ViT's final feature map. Scale 1/8 (or 1/4) is 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 LN and LeakyReLU6 [31]. Then for each pyramid level, we apply a 1×1 convolution with LN to reduce 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 the robustness to 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.

To align training and inference processes, we apply a so-called STE for the LeakyReLU6 function [31]. In the forward pass, hard-bounded activation $y_{hard} = \min(\max(ax, x, c))$ with $c = 6$ is used, while in the backward pass, the gradient is taken from the "soft" version with smooth bounding $y_{hard} = c * \tanh(leaky\_relu(x)/c)$.

The STE implementation is given as $y = y_{hard} + stopgrad(y_{soft} - y_{hard})$, which preserves quantization-friendly and range-stable forward operations and provides smooth, informative gradients without gradient masking effects.

Detection head, which is applied to each feature map, calculates confidence and bounding boxes for detected objects. The detection head consists of regression box subnetworks and a classifier subnetwork (Fig. 5) [30]. 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 different size and aspect ratio [30].

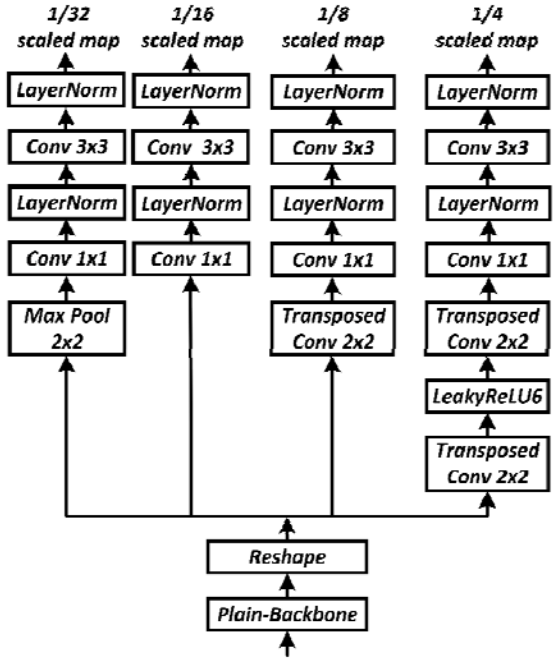

Figure 4 – Architecture of Simple Feature Piramid Network for Plain Backbone

Each target box is matched with anchor boxes at each step of the training. If the IoU between the anchor box and the 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 will be ignored during training. The classification subnetwork is trained with respect to the resulting assignments (object class or background). The regression subnetwork is trained with respect to the coordinates of the selected anchor box. The error is calculated relative to the anchor box, not the target box.

The training procedure is proposed to proceed in several stages (Fig. 6). In the first stage, we obtain an initialization that serves as a starting point for searching task-optimal parameters. Visual transformers can be pre-trained using any effective strategy; at present, the most capable self-supervised approaches include MAE, MSN [32], DINO [33], and DINOv2 [34]. In the second stage, a detection head is attached and the model is fine-tuned for object recognition in aerial imagery. Next, gate units and adapters are incorporated, and training is conducted on data with standard augmentation as well as with procedurally generated noise. Finally, when high predictive uncertainty is detected at the network output, test-time adaptation may be applied by tuning the adapters on augmented versions of the input to minimize the marginal entropy at the network's output.

DOI 10.15588/1607-3274-2026-2-8

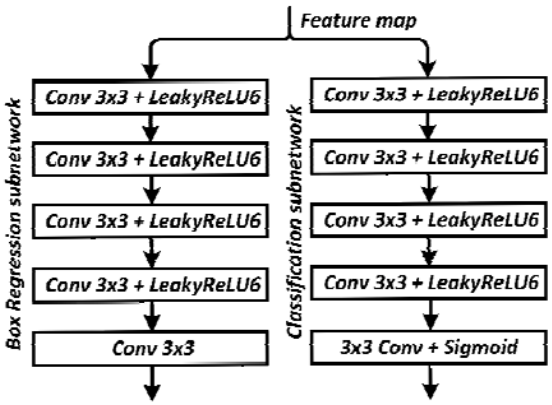

Figure 5 – RetinaNet-like Detection Head

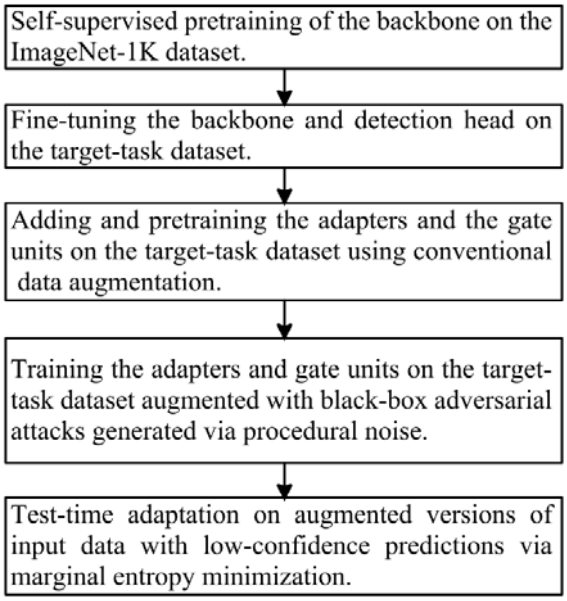

Figure 6 – Stages of the proposed training procedure for the object-detection neural network.

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

$$L = \lambda_{loc} L_{loc} + \lambda_{cls} L_{cls} + \lambda_{usage} L_{usage}. \quad (6)$$

Bounding box regression loss is calculated by the formula:

$$L_{loc} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j \in \{x,y,w,h\}} smooth_{L1}(P_{i,j} - T_{i,j}), \quad (7)$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, if |x| < 1, \\ |x| = 0.5, otherwise. \end{cases} \quad (8)$$

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

$$L_{cls} = -(1 - p_t)^{\gamma} \log(p_t). \quad (9)$$

Function (8) is an improved cross-entropy function. The difference consists in the addition of the parameter $\gamma \in (0,+\infty)$, which solves the problem of unbalanced classes. In training, most of the objects processed by the classifier are background, which is a separate class. Therefore, there may be a problem when the neural network learns to detect the background better than other objects. The additional parameter solved this problem by reducing the error value for easily classified object classes.

The L_{usage} function that calculates the deviation of the desired dynamic compression rate of the neural network from the real one. This function is affected by the number of activated blocks of the main model based on decisions made by the gate units. The L_{usage} function is calculated similarly to [34]:

$$L_{usage} = \frac{1}{K} \sum_{k=1}^{K} \left(\frac{1}{n} \sum_{i=1}^{n} g_{k,i} - t_k \right)^2. \quad (10)$$

To model natural but adversarial visual distortions that typically degrade image to model natural adversarial visual distortions that degrade image matching performance, we introduce structured procedural noise during training. Specifically, Gabor, Perlin, and Worley noise patterns perturb input data to approximate real-world conditions such as shadows, surface textures, occlusions, and lighting inconsistencies. Unlike unstructured Gaussian noise, these patterns introduce semantically plausible perturbations mimicking natural variability, forcing the model to focus on stable, semantically meaningful features rather than texture-dependent or unstable keypoints.

Perlin noise models organic variations similar to lighting gradients or natural surfaces. It is generated by interpolating gradients at grid node points and is typically implemented recursively to create fractal behavior. Perlin noise is constructed as a fractal sum of smoothed co-frequencies in frequency space [35]:

$$S_{per}(x,y) = \sum_{n=1}^{\Omega} p\left(x \frac{2^{n-1}}{\lambda_x}, y \frac{2^{n-1}}{\lambda_y} \right). \quad (11)$$

To enhance visual contrast, researchers often apply a sinusoidal transformation:

$$G_{per}(x,y) = \sin(2\pi \cdot \phi_{sine} \cdot S_{per}(x,y)). \quad (12)$$

DOI 10.15588/1607-3274-2026-2-8

Gabor noise imitates directional structures, such as cast shadows or repetitive patterns (e.g., fences, roofing). Gabor noise is defined as the convolution of sparse white noise with a Gabor kernel [9]:

$$g(x,y) = e^{-\pi\sigma^2(x^2+y^2)} \cdot \cos(\frac{2\pi}{\lambda}(x \cdot \cos(\omega) + \\ + y \cdot \sin(\omega)). \quad (13)$$

The resulting Gabor noise is generated as a sum of convolutions:

$$S_{gab}(x,y) = \frac{1}{\xi}\sum_{n=1}^{\xi}\sum_{i} g(x-x_i, y-y_i; \sigma, \lambda, \omega + \frac{n\pi}{\xi}). \quad (14)$$

Worley noise (cellular noise) models' structural irregularities, such as stone patterns, cracked surfaces, or uneven terrain. It is defined through distance computation to the nearest point in a grid of pseudo-random control points [35]:

$$W(x,y) = \min_{p_i \in F} \left\| (x,y) - p_i \right\|. \quad (15)$$

To ensure generalization during adversarial training, procedural noise parameters (orientation, frequency, phase, element spacing) are randomized per iteration or mini-batch. Perturbation amplitude is limited to 3–8% of dynamic range to preserve visual plausibility. The resulting image is formed by additive noise superposition:

$$I_{noisy} = I + \in \cdot N(x,y). \quad (16)$$

This produces barely noticeable yet semantically destructive perturbations, ideal for structured adversarial training, improving model robustness to visual variability in real aerial images.

Tuning of the adapter parameters using the method of marginal entropy minimization with one test point [36] to increase the robustness of the model. For each augmentation $b = 1..B$ of the same test input, the model head (pre-NMS) produces class logits $z_a^{(b)} \in R^C$ for each anchor a and an objectness logit $o_a^{(b)} \in R$. Averaging across augmentations is used to obtain the marginal class distribution, given by:

$$\overline{p}_a = \frac{1}{B}\sum_{i=1}^{B} p_a^{(b)}, \quad (17)$$

$$p_a^{(b)} = \text{softmax}(z_a^{(b)}). \quad (18)$$

At the pre-NMS level we have a huge number of anchor/grid cells $a = 1..A$, and most of them are background. If we minimize plain marginal entropy averaged over all cells, the loss is dominated by background cells. During TTA the model would then mainly learn to reduce entropy on background, diluting the useful signal from true object locations. Weighting the entropy by objectness focuses learning on cells that likely contain an object and suppresses background influence – conceptually similar to how focal loss downweights easy negatives. Therefore, the loss function for TTA takes the following form:

$$L_{TTA} = \frac{1}{A}\sum_{a=1}^{A} w_a H(\overline{p}_a) = \\ = -\frac{1}{A}\sum_{a=1}^{A} w_a \left(-\sum_{c=1}^{C} \overline{p}_{a,c} \log \overline{p}_{a,c} \right), \quad (19)$$

$$w_a = \left(\frac{1}{B}\sum_{i=1}^{B} \sigma(o_a^{(b)}) \right)^{\gamma}. \quad (20)$$

Accordingly, the proposed method enables model adaptation by dynamically activating the most relevant modules for processing the input data and by facilitating rapid adaptation to novelty through the application of a TTA mechanism to the adapters. Taken together, these components are aimed at improving object recognition performance in aerial imagery under challenging conditions.

4 EXPERIMENTS

Modern UAVs increasingly rely on companion computers to sustain autonomy under challenging conditions. We target representative mobile SoCs suitable for small UAVs: Rockchip RK3588 (designed in China, Manufactured in South Korea), MediaTek Dimensity 800 / 820 (designed in Taiwan, manufactured in Taiwan), Qualcomm Snapdragon 855 (designed in USA, manufactured in Taiwan). All of the cited chips feature a 4× Cortex-A76 + 4× Cortex-A55 big.LITTLE CPU cluster, which constitutes a high-performance configuration for mobile applications, offering strong single-threaded throughput from the A76 "big" cores and energy-efficient background processing on the A55 "little" cores.

To sustain >10 FPS on a CPU cluster comparable to 4× Cortex-A76 + 4× Cortex-A55, the end-to-end detector should not exceed ≈8–10 GFLOPs per image. Accordingly, we cap the backbone at ≤5–7 GFLOPs (for 512×512 inputs), leaving the remaining budget for the neck/head. Suitable backbones include ViT-T/16 (preferred) or ViT-S/16 with aggressive dynamic gating and/or a reduced input resolution [26]. To meet this budget we rely on dynamic compression via gate units, which skip computations in a data-dependent manner and

DOI 10.15588/1607-3274-2026-2-8

thus lower average FLOPs while preserving accuracy on difficult inputs. In this study we execute on CPU cores only, as this simplifies implementing dynamic control flow; current NPU toolchains generally favor static graphs and restricted operator sets, which complicates dynamic execution. Exploring NPU-based dynamic inference – e.g., via pre-deployed subnetworks and runtime selection – is left for future work.

This study does not explore the impact of different backbone pretraining methods on the efficiency of incorporating dynamic model compression or on robustness. Therefore, we will choose one of the most well-researched methods, MAE. In this case, the pretraining is done on the ImageNet-1k dataset with a resolution of 512x512 pixels [37]. During fine-tuning, a step-wise learning rate is used, starting at 0.1 and decaying by a factor of 10 after 150 and 250 epochs. VEDAI [38] and VisDrone [39] is a dataset of annotated images with a resolution of 512x512, used for supervised learning and test-time evaluation and adaptation.

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

The mAP for all detection categories is used as an evaluation metric of the object detector on aerial images. The Average Precision of each class is calculated as the area under the Precision-Recall curve [40]. Further, mAP is defined as mAP@0.5, which represents the mean Average Precision when the IoU threshold is set to 0.5.

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

5 RESULTS

To evaluate the effectiveness of the proposed approach, we analyze its training outcomes on the VEDAI dataset as an ablation study. Table 1 presents results obtained under training regimes with and without data augmentation via procedural noise. The table reports numerical values for mAP@0.5, the average computational cost in GFLOPs per frame, and processing throughput in FPS. All values are averaged over identical subsamples and are reported as most likely estimates for the target hardware; deviations of approximately ±0.5 FPS and ±0.01 mAP are possible due to data variability and the CPU scheduler.

Analysis of Table 1 indicates that, within the same domain (VEDAI), a dynamic backbone with compression ≈0.5 consistently halves the average FLOPs (from 9.6 GFLOPs per frame to ~5 GFLOPs per frame) and accelerates inference from 10 FPS to 17–18 FPS, while improving mAP by ≈+0.03–+0.04 over the conventional (static) counterpart. Pretraining on structured perturbations contributes an additional ≈+0.03 mAP, and incorporating lightweight adapters with TTA yields a further ≈+0.02–+0.03 mAP at a minor throughput penalty (~0.5–1.0 FPS). The highest accuracy is achieved by the configuration with a dynamic ViT-T/16 pretrained on data augmented with procedural noise (Perlin/Gabor/Worley) and using TTA (mAP 0.82), within the same computational budget. For VEDAI in the absence of domain shift, the dynamic backbone provides a better accuracy – throughput trade-off than the static one, while perturbation-informed pretraining and TTA deliver additional accuracy gains without exceeding the 8–10 GFLOPs budget.

Table 2 presents the evaluation results of the proposed model, trained on the VisDrone dataset, when assessed on the VEDAI dataset to simulate a domain shift. These experimental findings are intended to demonstrate how the model's behavior differs under domain shift.

Table 1 – Results of evaluating the proposed model on the VEDAI dataset after training on VEDAI

Training mode	Model and method	mAP@0.5	Avrg GFLOPs	FPS
Without pretraining on augmented data with procedural noise	Conventional ViT-T/16	0.74	9.6	10.6
	ViT-T/16 with trained gate unites (compression rate ≈ 0.5)	0.77	5.0	17.8
	ViT-T/16 with trained gate unites (compression rate ≈ 0.5) and adapters tuned by TTA	0.79	5.3	17.0
Pretraining on augmented data with procedural noise (Perlin/Gabor/Worley)	Conventional ViT-T/16	0.77	9.8	10.4
	ViT-T/16 with trained gate unites (compression rate ≈ 0.5)	0.80	5.2	17.3
	ViT-T/16 with trained gate unites (compression rate ≈ 0.5) and adapters tuned by TTA	0.82	5.5	16.6

DOI 10.15588/1607-3274-2026-2-8

Table 2 – Evaluation results of the proposed model on the VEDAI dataset, trained on unperturbed data from the VisDrone dataset

Training mode	Model and method	mAP@0.5	Avrg GFLOPs	FPS
Without pretraining on augmented data with procedural noise	Conventional ViT-T/16	0.54	9.6	10.4
	ViT-T/16 with trained gate unites (compression rate ≈ 0.5)	0.57	5.1	17.6
	ViT-T/16 with trained gate unites (compression rate ≈ 0.5) and adapters tuned by TTA	0.60	5.4	16.8
Pretraining on augmented data with procedural noise (Perlin/Gabor/Worley)	Conventional ViT-T/16	0.58	9.8	10.2
	ViT-T/16 with trained gate unites (compression rate ≈ 0.5)	0.62	5.3	17.1
	ViT-T/16 with trained gate unites (compression rate ≈ 0.5) and adapters tuned by TTA	0.66	5.6	16.3

Analysis of Tables 1 and 2 shows that due to domain shift (VisDrone→VEDAI), absolute metrics decrease compared to the single-domain setting (VEDAI→VEDAI): for the static version, mAP@0.5 drops from 0.74–0.77 to 0.54–0.58, and for the dynamic version with adapters and TTA from 0.79–0.82 to 0.60–0.66. However, the relative gain of the dynamic model is preserved at +0.03 to +0.08 mAP under a similar budget – approximately 5.4–5.6 GFLOPs for the dynamic model versus 9.6–9.8 GFLOPs for the static one – which confirms the better generalization ability of the proposed approach even under domain shift. At the same time, pretraining on perturbed data with procedural noise (Perlin/Gabor/Worley) consistently adds about +0.03 mAP to accuracy both within a single domain and under domain shift.

6 DISCUSSION

The results in Tables 1–2 demonstrate that a dynamic backbone with gate units reduces the average inference cost of a transformer detector to the level of smaller static models while preserving – or even improving – accuracy. The gates add a negligible fixed overhead (~0.015% FLOPs), yet under a target compression rate of ≈0.5–0.6 they skip non-essential encoder blocks on easy inputs, yielding ≈45–50% lower FLOPs on average without harming performance. In cross-domain transfer (VEDAI→VisDrone), this translates into 16–18 FPS on 4×A76 CPU within an 8–10 GFLOPs budget, with mAP rising from 0.54→0.60 without perturbation pretraining and 0.58→0.66 with it when light TTA is enabled (Tables 2). In the in-domain case (VEDAI→VEDAI), the same mechanism delivers 0.79–0.82 mAP at ~5–5.5 GFLOPs for the backbone and 17–18 FPS, again within the same total budget (Table 3).

Training under structured perturbations (Perlin/Gabor/Worley) consistently improves robustness: across both transfer and in-domain settings we observe +0.03…+0.05 mAP versus training on clean data. A lightweight test-time adaptation (entropy-minimization on adapters with objectness weighting) adds a further ≈+0.02…+0.04 mAP at a modest runtime cost (~0.5–1.3 FPS). Together, perturbation-aware training and TTA complement dynamic execution: gates learn to route around vulnerable or irrelevant subpaths, whereas adapters provide on-the-fly calibration under shift.

Placing these findings in the context of VEDAI benchmarks from the literature, heavier models do not guarantee better accuracy–compute trade-offs. For example, YOLOv9-t (11.1 GFLOPs, mAP50 0.654) underperforms YOLOv8-n (8.1 GFLOPs, mAP50 0.712); RT-DETR (103.6 GFLOPs, mAP50 0.455), TPH-YOLO (270.9 GFLOPs, mAP50 0.584), and Fast-RCNN (196.2 GFLOPs, mAP50 0.459) also show higher FLOPs with lower accuracy than lighter one-stage baselines [41]. In contrast, our dynamic ViT-T/16 matches or exceeds the accuracy of comparable small models while cutting average compute, and it maintains this advantage under domain shift and synthetic structured noise. These observations reinforce the core claim: “more FLOPs” ≠ “higher mAP” – especially in aerial detection where small-object density, texture confounds, and shift sensitivity are pronounced.

From a systems perspective, static compression techniques (quantization, pruning, distillation) remain attractive for wide hardware support because they produce fixed graphs. Dynamic networks, by design, rely on conditional control flow that is natively supported on CPU/GPU in PyTorch/TensorFlow and, as our measurements show, already pays off on CPU-class 4×A76. However, many current NPUs still favor static graphs and limited operator sets, complicating deployment of fully dynamic execution. A pragmatic path forward is hybridization: retain dynamic routing on CPU/GPU or via pre-materialized subnetworks selectable at runtime; combine with light static compression to widen deployability; bound TTA updates by strict budget and stability safeguards to preserve real-time throughput.

Dynamic gating plus perturbation-aware training and budgeted TTA constitute a compute-aware robustness stack for UAV object detection. This stack reduces average FLOPs, sustains or improves mAP under realistic shifts and disturbances, and meets embedded latency targets on commodity mobile CPUs – without requiring specialized accelerators.

CONCLUSIONS

This work presents an adaptive object detector for UAV aerial imagery that combines a dynamically gated ViT-T/16 backbone with parameter-efficient adapters, a Simple FPN, and a RetinaNet-like head, trained with procedural noise and supported by objectness-weighted test-time adaptation to improve robustness and efficiency

DOI 10.15588/1607-3274-2026-2-8

under onboard constraints. Ablations on VEDAI demonstrate a superior accuracy-throughput trade-off – approximately halving average FLOPs while increasing FPS and improving mAP – enabled by budget-aware dynamic execution, perturbation-aware training, and lightweight adaptation on the edge.

Limitations: Evaluation is centered on VEDAI at 512×512 and mAP@0.5, limiting conclusions on fine-grained localization and broader generalization. Experiments are CPU-only; compatibility, energy, and thermal behavior on NPUs/ASICs with dynamic control flow were not assessed. Pretraining is restricted to MAE on ImageNet-1k; interactions of alternative pretraining and adapters with dynamic routing and TTA remain unexplored. Procedural noise serves as a proxy for real disturbances; comprehensive field trials under diverse operational conditions are pending.

Future work will be connected with advancing hardware-software co-design by integrating dynamic execution policies with NPUs and specialized SoCs through pre-deployed subnetworks and intelligent runtime routing to preserve computational flexibility while leveraging hardware acceleration, which will necessitate comprehensive energy and thermal profiling under real-time constraints.

ACKNOWLEDGMENTS

The authors thank the Ministry of Education and Science of Ukraine for the support to the Laboratory of Intellectual Systems in the framework of research project No. 0124U000548 “Information Technology for Ensuring the Resilience of the Intelligent Onboard System of Small-Sized Aircraft” (2024–2026) та No. 0121U112684 “Implementation of the Tasks of the Prospective Development Plan for the Scientific Field ‘Technical Sciences’ of Sumy State University” (2025).

DECLARATIONS

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

Authors’ contributions: development of conceptual provisions and methodology of research, formulation of conclusions – A. S. Moskalenko, V. V. Moskalenko; review and analysis of references; development of software for modeling – A. Vatsenko; development of mathematical models, analysis of research results – Y. V. Moskalenko. All authors have read and agreed with the published version of the manuscript.

Data availability: This study uses publicly available datasets. For pretraining, we adopt MAE on the ImageNet-1k dataset [37] (access via the official ImageNet website: https://image-net.org/challenges/LSVRC/2012/; accessed on 24 October 2025). For supervised training, evaluation, and test-time adaptation, we use VEDAI (Vehicle Detection in Aerial Imagery) [38] (https://downloads.greyc.fr/vedai/; accessed on 24 October 2025) and VisDrone [39] (https://www.visdrone.org/dataset/; accessed on 24 October 2025). All datasets are redistributed by their respective maintainers under their original terms of use.

Software availability: The manuscript has no associated software.

Use of artificial intelligence: the authors utilized artificial intelligence technologies to provide their verified results. The writing of the article’s text was done without the use of artificial intelligence technologies.

REFERENCES

1. 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, № 1, P. 149. DOI: 10.3390/rs16010149.
2. Wei H., Wang Z., Ni Y. Hierarchical mixed-precision post-training quantization for SAR ship detection networks. *Remote Sensing,* 2024, Vol. 16, № 21, P. 4042. DOI: 10.3390/rs16214042.
3. Hendrycks D., Dietterich T. Benchmarking neural network robustness to common corruptions and perturbations [Electronic resource], 2019. Access mode: https://arxiv.org/abs/1903.12261. DOI: 10.48550/arXiv.1903.12261.
4. He Y., Meng G., Chen K., Hu X., He J. Towards security threats of deep learning systems: a survey. *IEEE Transactions on Software Engineering,* 2020, Vol. 48, № 5, pp. 1743 – 1770. DOI: 10.1109/TSE.2020.3034721.
5. Arkin E., Yadikar N., Xu X., Aysa A., Ubul K. A survey: object detection methods from CNN to transformer. *Multimedia Tools and Applications*, 2023, Vol. 82, № 14, pp. 21353–21383. DOI: 10.1007/s11042-022-13801-3.
6. Rahmath P. H., Srivastava V., Chaurasia K., Pacheco R. G., Couto R. S. Early-Exit deep neural network – a comprehensive survey. *ACM Computing Surveys,* 2024, Vol. 57, № 3, pp. 1–37. DOI: 10.1145/3698767.
7. Sun Y., Sun Z., Chen W. The evolution of object detection methods, *Engineering Applications of Artificial Intelligence,* 2024, Vol. 133, № 108458. DOI: 10.1016/j.engappai.2024.108458.
8. Cao J., Peng B., Gao M., Hao H., Li X., Mou H. Object detection based on CNN and vision-transformer: a survey. *IET Computer Vision*, 2025, Vol. 19, № 1, pp. 1–30. DOI: 10.1049/cvi2.70028.
9. Papa L., Russo P., Amerini I., Zhou L. A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2024, Vol. 46, № 12, pp. 7682–7700. DOI: 10.1109/TPAMI.2024.3392941.
10. Ruan X., Tang W. Fully test time adaptation for object detection, *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, USA, 17–18 June 2024 : proceedings*. Piscataway, IEEE, 2024, pp. 1038–1047. DOI: 10.1109/CVPRW63382.2024.00110.
11. Li Y., Fan Q., Huang H., Han Z., Gu Q. A modified YOLOv8 detection network for UAV aerial image recognition. *Drones,* 2023, Vol. 7, № 5. DOI: 10.3390/drones7050304.

DOI 10.15588/1607-3274-2026-2-8

12. 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, № 6, P. 261. DOI: 10.3390/drones8060261.
13. Lyu Z., Yu T., Pan F., Zhang Y., Luo J., Zhang D., Chen Y., Zhang B., Li G. A survey of model compression strategies for object detection. *Multimedia Tools and Applications,* 2023, Vol. 83, P. 48165–48236. DOI: 10.1007/s11042-023-17192-x.
14. Ju W., Bao W., Ge L., Yuan D. Dynamic early exit scheduling for deep neural network inference through contextual bandits. *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, New York, NY, USA, 1–5 November 2021 : proceedings*. New York, ACM, 2021, pp. 823–832. DOI: 10.1145/3459637.3482335.
15. Yin H., Vahdat A., Alvarez, J. M. Mallya A., Kautz J., Molchanov P. A-ViT: adaptive tokens for efficient vision transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022 : proceedings*. Piscataway, IEEE, 2022. DOI: 10.48550/arXiv.2112.07658.
16. Li Y., Xie B., Guo S., Yang Y., Xiao B. A survey of robustness and safety of 2D and 3D deep learning models against adversarial attacks, *ACM Computing Surveys,* 2024, Vol. 56, № 6, pp. 1–37. DOI: 10.1145/3636551.
17. Awad Z., Zakaria M., Hassan R. An enhanced ensemble defense framework for boosting adversarial robustness of intrusion detection systems. *Scientific Reports*, 2025, Vol. 15, № 1, P. 94023. DOI: 10.1038/s41598-025-94023-z.
18. Chen Y., Shen Y., Duan C., Wang Z., Mo Z., Liang Y., Zhang Q. Robust and efficient SAR ship detection: an integrated despecking and detection framework. *Remote Sensing*, 2025, Vol. 17, № 4, P. 580. DOI: 10.3390/rs17040580.
19. 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), Paris, France, 2–6 October 2023 : proceedings*. Piscataway : IEEE, 2023. DOI: 10.1109/ICCVW60793.2023.00163.
20. Liu J., Jin Y. A comprehensive survey of robust deep learning in computer vision. *Journal of Automation and Intelligence*, 2023, Vol. 2, № 4, pp. 175–195. DOI: 10.1016/j.jai.2023.10.002.
21. Wang S., Veldhuis R., Brune C., Strisciuglio N. A survey on the robustness of computer vision models against common corruptions. [Electronic resource], 2023. Access mode: https://arxiv.org/abs/2305.06024.
22. Gharoun H., Momenifar F., Chen F., Gandomi A. H.Meta-learning approaches for few-shot learning: a survey of recent advances. *ACM Computing Surveys,* 2024, Vol. 57, № 8. DOI: 10.1145/3659943.
23. Rao Y., Liu Z., Zhao W., Zhou J., Lu J. Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2023, pp. 1–14. DOI: 10.48550/arXiv.2106.02034.
24. Wang D., Shelhamer E., Liu S., Olshausen B., Darrell T. Tent: fully test-time adaptation by entropy minimization. [Electronic resource], 2021. Access mode: https://arxiv.org/abs/2006.10725. DOI: 10.48550/arXiv.2006.10725.
25. Maesumi A., Hu D., Saripalli K., Kim V. G., Fisher M., Pirk S., Ritchie D. One noise to rule them all: learning a unified model of spatially-varying noise patterns. *ACM Transactions on Graphics,* 2024, Vol. 43, № 4, pp. 1–21. DOI: 10.1145/3658195.
26. Touvron H., Cord M., Douze M., Massa F., Sablayrolles A., Jégou H. Training data-efficient image transformers & distillation through attention. [Electronic resource], 2020. Access mode: https://arxiv.org/abs/2012.12877. DOI: 10.48550/arXiv.2012.12877.
27. Scardapane S., Baiocchi A., Devoto A., Marsocci V., Minervini P., Pomponi J. Conditional computation in neural networks: principles and research trends. *Intelligenza Artificiale,* 2024, Vol. 18, № 1. DOI: 10.3233/IA-240035.
28. Meng L., Li H., Chen B., Lan S., Wu Z., Jiang Y., S. Lim AdaViT: adaptive vision transformers for efficient image recognition. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA, 18–24 June 2022 : proceedings*. Piscataway, IEEE, 2022. DOI: 10.1109/CVPR52688.2022.01199.
29. Chen S., Ge C., Tong Z., Wang J., Song Y., Wang J., Luo P. AdaptFormer: adapting vision transformers for scalable visual recognition. *NeurIPS: Conference on Neural Information Processing Systems*, 2022 : proceedings, 2022. DOI: 10.5555/3600270.3601482.
30. Li Y., Mao H., Girshick R., He K. Exploring plain vision transformer backbones for object detection. *Lecture Notes in Computer Science.* Cham, Springer, 2022, pp. 280–296. DOI: 10.1007/978-3-031-20077-9_17.
31. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L. MobileNetV2: inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018 : proceedings*. Piscataway, IEEE, 2018. DOI: 10.1109/CVPR.2018.00474.
32. 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*. Cham, Springer, 2022, pp. 456–473. DOI: 10.1007/978-3-031-19821-2_26.
33. Caron M., Touvron H., Misra I., Jégou H., Mairal J., Bojanowski P., Joulin A. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021 : proceedings*. Piscataway, IEEE, 2021. DOI: 10.1109/ICCV48922.2021.00951.
34. Oquab M., Darcet T., Moutakanni T., Vo H., Szafraniec M., Khalidov V., Fernandez P., Haziza D., Massa F., El-Nouby A., Assran M., Ballas N., Galuba W., Howes R., Huang P., Li S., Misra I., Rabbat M., Sharma V., Synnaeve G., Xu H., Jegou H., Mairal J., Labatut P., Joulin A., Bojanowski P. DINOv2: learning robust visual features without supervision. [Electronic resource], 2023. Access mode: https://arxiv.org/abs/2304.07193.
35. Lagae A., Lefebvre S., Cook R., Derose T., Drettakis G., Ebert D. S., Lewis J. P., Perlin K., Zwicker M. A survey of procedural noise functions. *Computer Graphics Forum*, 2010, Vol. 29, № 8, pp. 2579–2600. DOI: 10.1111/j.1467-8659.2010.01827.x.
36. Zhang M., Levine S., Finn C. MEMO: test time robustness via adaptation and augmentation [Electronic resource], 2021. Access mode: https://arxiv.org/abs/2106.07596.
37. He K., Chen X., Xie S., Li Y., Dollar P., Girshick R. Masked autoencoders are scalable vision learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, USA, 18–24 June 2022 :*

DOI 10.15588/1607-3274-2026-2-8

proceedings. Piscataway, NJ, IEEE, 2022. DOI: 10.1109/CVPR52688.2022.01553.
38. Tariq J., Kwong S., Yuan H. HEVC intra mode selection based on rate-distortion (RD) cost and sum of absolute difference (SAD). *Journal of Visual Communication and Image Representation,* 2016, Vol. 35, pp. 112–119. DOI: 10.1016/j.jvcir.2015.11.013.
39. Cao Y., He Z., Wang L., Wang W., Yuan Y., Zhang D., Zhang J., Zhu P., Gool L. V., Han J., Hoi S., Hu1 Q., Liu M., Cheng C., Liu F., Cao G., Li G., Wang H., He J., Wan J., Wan Q., Zhao Q., Lyu S., Zhao W., Lu X., Zhu X., Liu Y., Lv Y., Ma Y., Yang Y., Wang Z., Xu Z., Luo Z., Zhang Z., Zhang Z., Li Z., Zhang Z. VisDrone DET2021: the vision meets drone object detection challenge results. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). Montreal, BC, Canada, 11–17 October 2021 : proceedings*. Piscataway, NJ, IEEE, 2021. DOI: 10.1109/ICCVW54120.2021.00319.
40. Leng J., Ye Y., Mo M., Gao C., Gan J., Xiao B., Gao X. Recent advances for aerial object detection: a survey. *ACM Computing Surveys*, 2024, Vol. 56, № 12. DOI: 10.1145/3664598.
41. Qiu Y., Zheng X., Hao X., Zhang G., Lei T., Jiang P. ARSOD-YOLO: enhancing small target detection for remote sensing images. *Sensors,* 2024, Vol. 24, № 23, P. 7472. DOI: 10.3390/s24237472.

Received 08.01.2026.
Accepted 20.04.2026.
Published 26.06.2026.

УДК 004.891.032.26:629.7.01.066

РЕСУРСОЕФЕКТИВНЕ, АДАПТИВНЕ ТА НАДІЙНЕ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ НА АЕРОЗНІМКАХ З БПЛА

Москаленко В. В. – канд. техн. наук, доцент, доцент кафедри комп'ютерних наук, Сумський державний університет, Суми, Україна. ROR: https://ror.org/01w60n236. ORCID: https://orcid.org/0000-0001-6275-9803.

Москаленко А. С. – канд. техн. наук,, доцент, доцент кафедри комп'ютерних наук, Сумський державний університет, Суми, Україна. ROR: https://ror.org/01w60n236. ORCID: https://orcid.org/0000-0003-3443-3990.

Москаленко Ю. В. – аспірант, Сумський державний університет, Суми, Україна. ROR: https://ror.org/01w60n236. ORCID: https://orcid.org/0000-0002-9121-7832.

Ваценко А. В. – аспірант, Сумський державний університет, Суми, Україна. ROR: https://ror.org/01w60n236. ORCID: https://orcid.org/0009-0007-3278-0194.

АНОТАЦІЯ

Актуальність. Забезпечення надійного, адаптивного та обчислювально ефективного детектування об'єктів на аерофотознімках БПЛА за умов зміщення розподілу, структурованого та неструктурованого шуму, а також суворих ресурсних і часових обмежень на борту БПЛА є актуальним науковим завданням.

Предмети дослідження. Модель детектора з урахуванням обчислювальної складності та метод навчання і адаптації, що інтегрують трансформаторний екстрактор ознак із динамічними вентилями, ефективні адаптери та адаптацію під час тестування в умовах обмежених ресурсів та впливу реалістичних збурень або зсуву домену для забезпечення точності рішень.

Мета. Розроблення моделі та методу детектування об'єктів на аерофотознімках, які спільно забезпечують робастність та адаптивність, водночас задовольняючи обчислювальні та часові обмеження, типові для бортових систем БПЛА.

Метод. Підхід поєднує динамічні нейронні мережі на основі екстракторі ознак ViT-T/16 з вентелями Гумбель-Softmax, просту мережу піраміди ознак (FPN) та одноступеневу детектуючу головку типу RetinaNet, функцію втрат з урахуванням бажаного коефіцієнта динамічного стиснення, структурований процедурний шум (Perlin, Gabor, Worley) для підвищення робастності навчання, функцію активації LeakyReLU6 з прямою оцінкою для стабільних градієнтів, а також метод адаптації під час тестування на основі мінімізації зваженої маржинальної ентропії на ефективних адаптерах.

Результати. На VEDAI, динамічний детектор на основі ViT T/16 за валідаційною метрикою mAP@0.5 досягає значення 0,77 з середньою кількістю обчислень ~5.0 Гігафлопс на кадр та швидкодією всередньому 17,8 кадрів/секунда і зростає за валідаційною метрикою до 0,79 з адаптерами та адаптацією під час тестування, тоді як статичний аналог досягає 0,74 при 9,6 Гігафлопс та 10,6 кадрів/секунду; попереднє навчання з процедурним шумом підвищує точність до 0,80 (динамічні вентелі) та 0,82 (динамічні вентелі + адаптація під час тестування) з мінімальними обчислювальними витратами. При зсуві домену (навчено на VisDrone, оцінено на VEDAI), динамічні вентелі та адаптація під час тестування покращують mAP@0.5 з 0,54 до 0,60 без попереднього навчання з шумом та до 0,66 з ним, підтримуючи ~5,4–5,6 Гігафлопс та ~16–17 кадрів/секунда в рамках бюджету 8–10 Гігафлопс на процесорах 4×A76.

Висновки. Запропонована модель та метод детектування об'єктів, що поєднують динамічні вентелі, навчання з урахуванням збурень та адаптацію під час тестування, зменшують середню кількість обчислення, одночасно підвищуючи робастність та адаптивність, що забезпечує кращий компроміс між точністю та пропускною здатністю для розгортання на борту БПЛА в умовах реальних збурень та змін розподілу.

КЛЮЧОВІ СЛОВА: детектування об'єктів, робастність, адаптивність, змагальний процедурний шум, динамічна нейронна мережа.

DOI 10.15588/1607-3274-2026-2-8

ЛІТЕРАТУРА

1. A survey of object detection for UAVs based on deep learning / [G. Tang, J. Ni, Y. Zhao et al.] // Remote Sensing. – 2023. – Vol. 16, № 1. – P. 149. DOI: 10.3390/rs16010149.
2. Wei H. Hierarchical mixed-precision post-training quantization for SAR ship detection networks / H. Wei, Z. Wang, Y. Ni // Remote Sensing. – 2024. – Vol. 16, № 21. – P. 4042. DOI: 10.3390/rs16214042.
3. Hendrycks D. Benchmarking neural network robustness to common corruptions and perturbations / D. Hendrycks, T. Dietterich [Electronic resource]. – 2019. – Access mode: https://arxiv.org/abs/1903.12261. DOI: 10.48550/arXiv.1903.12261.
4. Towards security threats of deep learning systems: a survey / [Y. He, G. Meng, K. Chen et al.] // IEEE Transactions on Software Engineering. – 2020. – Vol. 48, № 5. – P. 1743 – 1770. DOI: 10.1109/TSE.2020.3034721.
5. A survey: object detection methods from CNN to transformer / [E. Arkin, N. Yadikar, X. Xu et al.] // Multimedia Tools and Applications. – 2023. – Vol. 82, № 14. – P. 21353–21383. DOI: 10.1007/s11042-022-13801-3.
6. Early-Exit deep neural network – a comprehensive survey / [P. H. Rahmath, V. Srivastava, K. Chaurasia et al.] // ACM Computing Surveys. – 2024. – Vol. 57, № 3. – P. 1–37. DOI: 10.1145/3698767.
7. Sun Y. The evolution of object detection methods / Y. Sun, Z. Sun, W. Chen // Engineering Applications of Artificial Intelligence. – 2024. – Vol. 133. – № 108458. DOI: 10.1016/j.engappai.2024.108458.
8. Object detection based on CNN and vision-transformer: a survey / [J. Cao, B. Peng, M. Gao et al.] // IET Computer Vision. – 2025. – Vol. 19, № 1. – P. 1–30. DOI: 10.1049/cvi2.70028.
9. A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking / [L. Papa, P. Russo, I. Amerini, L. Zhou] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2024. – Vol. 46, № 12. – P. 7682–7700. DOI: 10.1109/TPAMI.2024.3392941.
10. Ruan X. Fully test time adaptation for object detection / X. Ruan, W. Tang // 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, USA, 17–18 June 2024 : proceedings. – Piscataway : IEEE, 2024. – P. 1038–1047. DOI: 10.1109/CVPRW63382.2024.00110.
11. A modified YOLOv8 detection network for UAV aerial image recognition / [Y. Li, Q. Fan, H. Huang et al.] // Drones. – 2023. – Vol. 7, № 5. DOI: 10.3390/drones7050304.
12. EUAVDet: an efficient and lightweight object detector for UAV aerial images with an edge-based computing platform / [W. Wu, A. Liu, J. Hu et al.] // Drones. – 2024. – Vol. 8, № 6. – P. 261. DOI: 10.3390/drones8060261.
13. A survey of model compression strategies for object detection / [Z. Lyu, T. Yu, F. Pan et al.] // Multimedia Tools and Applications. – 2023. – Vol. 83. – P. 48165–48236. DOI: 10.1007/s11042-023-17192-x.
14. Dynamic early exit scheduling for deep neural network inference through contextual bandits / [W. Ju, W. Bao, L. Ge, D. Yuan] // CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia. – New York, NY, USA, 1–5 November 2021 : proceedings. – New York : ACM, 2021. – P. 823–832. DOI: 10.1145/3459637.3482335.
15. A-ViT: adaptive tokens for efficient vision transformer / [H. Yin, A. Vahdat, J. M. Alvarez et al.] // 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022 : proceedings. – Piscataway : IEEE, 2022. DOI: 10.48550/arXiv.2112.07658.
16. A survey of robustness and safety of 2D and 3D deep learning models against adversarial attacks / [Y. Li, B. Xie, S. Guo et al.] // ACM Computing Surveys. – 2024. – Vol. 56, № 6. – P. 1–37. DOI: 10.1145/3636551.
17. Awad Z. An enhanced ensemble defense framework for boosting adversarial robustness of intrusion detection systems / Z. Awad, M. Zakaria, R. Hassan // Scientific Reports. – 2025. – Vol. 15, № 1. – P. 94023. DOI: 10.1038/s41598-025-94023-z.
18. Robust and efficient SAR ship detection: an integrated despecking and detection framework / [Y. Chen, Y. Shen, C. Duan et al.] // Remote Sensing. – 2025. – Vol. 17, № 4. – P. 580. DOI: 10.3390/rs17040580.
19. Haque M. Dynamic neural network is all you need: understanding the robustness of dynamic mechanisms in neural networks / M. Haque, W. Yang // 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Paris, France, 2–6 October 2023 : proceedings. – Piscataway : IEEE, 2023. DOI: 10.1109/ICCVW60793.2023.00163.
20. Liu J. A comprehensive survey of robust deep learning in computer vision / J. Liu, Y. Jin // Journal of Automation and Intelligence. – 2023. – Vol. 2, № 4 – P. 175–195. DOI: 10.1016/j.jai.2023.10.002.
21. A survey on the robustness of computer vision models against common corruptions / [S. Wang, R. Veldhuis, C. Brune, N. Strisciuglio] [Electronic resource]. – 2023. – Access mode: https://arxiv.org/abs/2305.06024.
22. Meta-learning approaches for few-shot learning: a survey of recent advances / [H. Gharoun, F. Momenifar, F. Chen, A. H. Gandomi] // ACM Computing Surveys. – 2024. – Vol. 57, № 8. DOI: 10.1145/3659943.
23. Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks / [Y. Rao, Z. Liu, W. Zhao, J. Zhou, J. Lu] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2023. – P. 1–14. DOI: 10.48550/arXiv.2106.02034.
24. Tent: fully test-time adaptation by entropy minimization / [D. Wang, E. Shelhamer, S. Liu et al.] // [Electronic resource]. – 2021. – Access mode: https://arxiv.org/abs/2006.10725. DOI: 10.48550/arXiv.2006.10725.
25. One noise to rule them all: learning a unified model of spatially-varying noise patterns / [A. Maesumi, D. Hu, K. Saripalli et al.] // ACM Transactions on Graphics. – 2024. – Vol. 43, № 4. – P. 1–21. DOI: 10.1145/3658195.
26. Touvron H. Training data-efficient image transformers & distillation through attention / [H. Touvron, M. Cord, M. Douze et al.] [Electronic resource]. – 2020. – Access mode: https://arxiv.org/abs/2012.12877. DOI: 10.48550/arXiv.2012.12877.
27. Conditional computation in neural networks: principles and research trends / [S. Scardapane, A. Baiocchi, A. Devoto et al.] // Intelligenza Artificiale. – 2024. – Vol. 18, № 1. DOI: 10.3233/IA-240035.
28. AdaViT: adaptive vision transformers for efficient image recognition / [L. Meng, H. Li, B. Chen et al.] // 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June

DOI 10.15588/1607-3274-2026-2-8

2022 : proceedings. – Piscataway : IEEE, 2022. DOI: 10.1109/CVPR52688.2022.01199.
29. AdaptFormer: adapting vision transformers for scalable visual recognition / [S. Chen, C. Ge, Z. Tong et al.] // NeurIPS: Conference on Neural Information Processing Systems, 2022 : proceedings. – 2022. DOI: 10.5555/3600270.3601482.
30. Exploring plain vision transformer backbones for object detection / [Y. Li, H. Mao, R. Girshick, K. He] // Lecture Notes in Computer Science. – Cham : Springer, 2022. – P. 280–296. DOI: 10.1007/978-3-031-20077-9_17.
31. MobileNetV2: inverted residuals and linear bottlenecks / [M. Sandler, A. Howard, M. Zhu et al.] // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018 : proceedings. – Piscataway : IEEE, 2018. DOI: 10.1109/CVPR.2018.00474.
32. Masked siamese networks for label-efficient learning / [M. Assran, M. Caron, I. Misra et al.] // Lecture Notes in Computer Science. – Cham: Springer, 2022. – P. 456–473. DOI: 10.1007/978-3-031-19821-2_26.
33. Emerging properties in self-supervised vision transformers / [M. Caron, H. Touvron, I. Misra et al.] // 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021 : proceedings. – Piscataway: IEEE, 2021. DOI: 10.1109/ICCV48922.2021.00951.
34. DINOv2: learning robust visual features without supervision / [M. Oquab, T. Darcet, T. Moutakanni et al.] [Electronic resource]. – 2023. – Access mode: https://arxiv.org/abs/2304.07193.
35. A survey of procedural noise functions / [A. Lagae, S. Lefebvre, R. Cook et al.] // Computer Graphics Forum. – 2010. – Vol. 29, № 8. – P. 2579–2600. DOI: 10.1111/j.1467-8659.2010.01827.x.
36. Zhang M. MEMO: test time robustness via adaptation and augmentation / M. Zhang, S. Levine, C. Finn [Electronic resource]. – 2021. – Access mode: https://arxiv.org/abs/2106.07596.
37. Masked autoencoders are scalable vision learners / [K. He, X. Chen, S. Xie et al.] // 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, USA, 18–24 June 2022 : proceedings. – Piscataway, NJ : IEEE, 2022. DOI: 10.1109/CVPR52688.2022.01553.
38. Tariq J. HEVC intra mode selection based on rate-distortion (RD) cost and sum of absolute difference (SAD) / J. Tariq, S. Kwong, H. Yuan // Journal of Visual Communication and Image Representation. – 2016. – Vol. 35. – P. 112–119. DOI: 10.1016/j.jvcir.2015.11.013.
39. VisDrone DET2021: the vision meets drone object detection challenge results / [Y. Cao, Z. He, L. Wang et al.] // 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021 : proceedings. – Piscataway, NJ : IEEE, 2021. DOI: 10.1109/ICCVW54120.2021.00319.
40. Recent advances for aerial object detection: a survey / [J. Leng, Y. Ye, M. Mo et al.] // ACM Computing Surveys. – 2024. – Vol. 56, № 12. DOI: 10.1145/3664598.
41. ARSOD-YOLO: enhancing small target detection for remote sensing images / [Y. Qiu, X. Zheng, X. Hao et al.] // Sensors. – 2024. – Vol. 24, № 23. – P. 7472. DOI: 10.3390/s24237472.

DOI 10.15588/1607-3274-2026-2-8