ISSN 3083-7715 (online) *Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189)*
ISSN 3083-7650 (0135-1710) (print) *Management Information System and Devices. 2026. No. 2 (189)*

UDC 004.032.26 DOI: https://doi.org/10.30837/0135-1710.2026.189.034

Stanislav Nechypurenko, Anton Sorokin

METHOD OF STRUCTURAL PRUNING OF CONVOLUTIONAL NEURAL NETWORKS WITH REGARD TO EDGE DEVICE HARDWARE CHARACTERISTICS

Research subject – methods of structural pruning of convolutional neural networks applied to reduce the computational complexity of object detection models in video systems with local data processing. **Purpose of the study** – improvement of the structural pruning method through the introduction of a hardware-aware channel selection criterion that takes into account the actual operation latency on the target edge device. **Research objectives**: comparative analysis of existing structural pruning criteria, justification for selecting the baseline method, and development of its modification based on latency profiling of the target hardware platform. **Research methods** are based on the mathematical analysis of convolutional layer channel importance criteria, theoretical comparison of approaches to evaluating neural network parameter redundancy, and formalization of the hardware-aware optimization problem. **Research results** consist in the development of a modified channel importance criterion that combines the evaluation of a channel's impact on the loss function with a normalized latency coefficient of the corresponding layer, calculated by profiling the execution time of each convolutional layer on the target edge device. The proposed criterion enables prioritization of channel removal in layers that impose the greatest computational load on specific hardware while preserving channels with a high impact on model accuracy. A theoretical analysis of the criterion properties was performed, including proof of its reduction to the baseline Taylor criterion when the hyperparameter value is zero, as well as the monotonicity of pruning priority redistribution between layers. It was demonstrated that the proposed approach is modular, requires minimal additional computational overhead, and can be integrated into existing model optimization pipelines. Experimental validation on the YOLOv8n detector model confirmed the effectiveness of the proposed approach: with a pruning budget of 30% removed channels, the inference latency was reduced by 38.55% compared to 34.88% for the baseline Taylor criterion, corresponding to a relative improvement of +10.5%. **Conclusions.** The results confirm the theoretical validity of the proposed approach and its advantages over hardware-independent pruning criteria for deploying detection models on edge devices as part of real-time video systems.

Keywords: structural pruning; convolutional neural network; edge device; inference latency; channel importance criterion; model optimization; video analytics; object detection; local data processing; hardware-aware optimization.

1. Introduction

The rapid growth in the number of surveillance cameras, transportation sensors, and IoT devices is driving the need for video analytics systems capable of operating in real time directly at the data collection site. Transmitting raw video streams to cloud servers results in significant consumption of communication channel bandwidth, increased latency, and risks of data privacy breaches [1]. An alternative approach is to deploy deep learning models directly on edge devices – local gateways, embedded processors, or specialized accelerators – which enables rapid decision-making and system autonomy even under unstable network conditions.

However, modern convolutional neural networks (CNNs), which demonstrate high accuracy in object detection and classification tasks, are characterized by significant demands on RAM, computational power, and energy consumption. These requirements significantly exceed the resource capabilities of typical edge platforms, such as ARM processors, NVIDIA Jetson, or Google Coral [2]. Overcoming this trade-off between model accuracy and computational complexity is one of the central challenges in the modern field of efficient deep learning.

Among the known approaches to reducing the computational complexity of CNNs are weight and activation quantization [3], knowledge distillation [4], neural architecture search [5], and structural pruning [6]. The latter approach, which involves removing entire channels or filters from convolutional layers, is attractive due to the direct reduction in the dimension of intermediate computation tensors without the need for specialized hardware for sparse arithmetic.

Existing criteria for structural pruning – based on the L1 norm [7], Taylor series expansion [8], and geometric median [9] – assess the importance of each channel solely in terms of its impact on model accuracy. This approach, however, does not account for the fact that reducing the number of floating-point operations (FLOPs) is not always proportional to the actual inference acceleration on specific hardware. Different network layers have varying execution efficiency depending on the processor architecture, cache size, memory access

Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189) *ISSN 3083-7715 (online)*
Management Information System and Devices. 2026. No. 2 (189) *ISSN 3083-7650 (0135-1710) (print)*

characteristics, and the degree of computational parallelism. As a result, a model optimized using the " " criterion – which aims for the minimum accuracy loss given a fixed number of removed parameters – may exhibit suboptimal acceleration on the target edge device.

This paper proposes an improvement to the structural pruning method based on the Taylor series expansion by introducing a hardware-oriented coefficient that accounts for the actual latency of operations in each layer on the target platform. The rationale for selecting the base method is based on a comparative analysis of existing pruning criteria.

2. Analysis of the literature and definition of the research problem

Compression methods for convolutional neural networks encompass a wide range of approaches that differ in their mechanisms for reducing computational complexity and their hardware requirements. Quantization involves transitioning from 32-bit floating-point representation to reduced-precision integer arithmetic. This approach provides up to a fourfold speedup provided there is hardware support for INT8 operations [3]. Knowledge distillation involves training a compact student model under the guidance of a more powerful teacher model, allowing a significant portion of accuracy to be preserved while substantially reducing the number of parameters [4]. Neural Architecture Search (NAS) automates the design of efficient network topologies but requires significant computational resources during the search phase [5].

Structural pruning occupies a unique position among these methods due to a number of practical advantages. Unlike non-structural pruning, which removes individual weights and creates sparse matrices, the structural approach removes entire channels or filters, resulting in a network that retains a dense tensor structure and can run efficiently on standard hardware without specialized libraries for sparse arithmetic [6]. Furthermore, structural pruning can be combined with other compression methods, notably quantization, providing a cumulative acceleration effect. Recent studies, notably [10], address the issue of inter-layer dependencies in structural pruning, enabling the correct removal of channels in networks with complex topologies.

Among the criteria for determining the importance of channels for structural pruning, the following approaches have been most extensively studied.

The criterion based on the L1 norm [7] evaluates the importance of the j-th filter in the l-th layer as the sum of the absolute values of its weights:

$$S_{L1}\left(f_j^l\right)=\sum_i w_i^{j,l}, \tag{1}$$

where $w_i^{j,l}$ is the i-th weight of the filter f_j^l in layer l. Filters with the smallest value of $S_{L1}\left(f_j^l\right)$ are considered redundant and are removed. The advantage of the method is computational simplicity – calculating the criterion does not require a forward or backward pass of the network. However, the $L1$ norm of the weights is only an indirect indicator of the channel's influence on the network's output signal and does not account for interactions between filters in different layers.

The criterion based on batch normalization scaling factors [11] uses the γ parameters of the BatchNorm layer as an indicator of the importance of the corresponding channel:

$$S_{BN}\left(c_j^l\right)=\left|\gamma_j^l\right|, \tag{2}$$

where γ_j^l is the batch normalization scaling coefficient for the j-th channel of the l-th layer. During training, $L1$ regularization of the parameters γ is added to the loss function, which encourages the network to set to zero the coefficients of channels that have the least impact on the result. This approach integrates the process of identifying redundant channels directly into the training procedure, but requires retraining the network with a modified loss function.

A criterion based on the Taylor series [8] evaluates the impact of channel removal on the loss function value through the first term of the Taylor series:

$$S_T\left(c_j^l\right)=\left|\frac{\partial L}{\partial c_j^l}\cdot c_j^l\right|, \tag{3}$$

where L is the loss function, and c_j^l is the activation of the j-th channel in the l-th layer. Unlike the $L1$ norm, this criterion takes into account not only the magnitude of the weights but also their actual impact on the loss function via gradient information. This provides a more accurate assessment of a channel's importance, especially in networks with complex architectures and residual connections. Computing the criterion requires one

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189)
ISSN 3083-7650 (0135-1710) (print) Management Information System and Devices. 2026. No. 2 (189)

forward and one backward pass on the training dataset, which is an acceptable computational cost.

The criterion based on the geometric median [9] is founded on a different idea: instead of estimating the absolute importance of each filter, its uniqueness relative to other filters in the same layer is determined:

$$S_{GM}\left(f_j^l\right) = \sum_{i \neq j} \left\| f_j^l - f_i^l \right\|_2, \quad (4)$$

where $\| \|_2$ is the Euclidean norm. The filters located closest to the geometric median of the set of filters in the layer are considered the most redundant, since their functionality can be replicated by other filters. The advantage of the method is its data independence – the criterion is calculated solely based on the network weights. However, the method does not account for the impact of filter removal on the network's overall accuracy in the context of a specific task. A comparative analysis of the considered criteria is presented in Table 1.

Table 1. *Comparison of CNN structural pruning criteria*

Criterion	What it evaluates	Data requirements	Advantage	Limitations
$L1$ norm	Absolute value of weights	Not required	Ease of calculation	Does not account for the impact on the loss function
BatchNorm	Scaling coefficients γ	Requires retraining	Integration with training	Requires modification of the loss function
Taylor	Impact on the loss function	Calibration sample	Accurate importance estimation	Requires a backpass
Geometric median	Filter uniqueness	Does not require	Data independence	Does not account for impact on accuracy

An analysis of the approaches presented reveals a common limitation: all the criteria considered evaluate the importance of channels solely from the perspective of their impact on model accuracy or the structural redundancy of the network. None of them takes into account the actual computational cost of executing operations of a specific layer on the target hardware platform.

This limitation becomes critical for edge deployment, since theoretical computational complexity in FLOPs is not an adequate predictor of actual inference latency. As noted in [2], layers with the same number of FLOPs can have significantly different latencies depending on the tensor dimensions (which affects cache efficiency), the type of operations (standard convolution versus depth-wise convolution), and the degree of parallelism of the available computational units. The AMC approach [12] partially addresses this issue by using reinforcement learning to determine optimal pruning coefficients for each layer, subject to latency constraints. However, AMC requires a lengthy agent training process and does not directly modify the channel importance criterion, which complicates its integration with existing pruning pipelines.

Thus, the task of developing a structural pruning criterion that combines an accurate assessment of a channel's impact on the loss function with consideration of the actual latency of the layer on the target edge device is of great relevance.

3. Research Objectives and Tasks

The objective of the research is to improve the method of structural pruning of convolutional neural networks for video systems with local data processing by introducing a hardware-oriented channel selection criterion that accounts for the actual latency of operations on the target edge device.

To achieve this goal, the following tasks have been defined:

- conduct a comparative analysis of existing structural pruning criteria and justify the selection of a baseline method for further improvement;
- develop a procedure for profiling the latency of CNN layers on the target hardware platform and formalize a hardware-oriented coefficient;
- formulate a modified channel importance criterion that combines the assessment of the impact on the loss function with the layer latency coefficient;
- perform a theoretical analysis of the properties of the proposed criterion and determine the conditions under which it provides advantages over the baseline method.

Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189) *ISSN 3083-7715 (online)*
Management Information System and Devices. 2026. No. 2 (189) *ISSN 3083-7650 (0135-1710) (print)*

4. Materials and Methods

4.1 Justification for the choice of the baseline method

Based on the comparative analysis performed in Section 2, the criterion based on Taylor's expansion was selected as the baseline method for improvement. This choice is justified by the following arguments:

- the Taylor criterion provides the most accurate assessment of the impact of channel removal on the loss function among the approaches considered, as confirmed by the results of comparative studies on standard architectures [8];
- unlike the $L1$ norm, the Taylor criterion takes gradient information into account, allowing channels with large but uninformative weights to be distinguished from channels with small but critically important weights;
- unlike the BatchNorm-based method [11], the Taylor criterion does not require retraining the network with a modified loss function, which simplifies its integration into the optimization pipeline;
- the computational cost of calculating the criterion (one forward and one backward pass on the calibration sample) is acceptable for practical application.

4.2 Latency Profiling Procedure

To obtain hardware-specific information about the computational cost of each network layer, we propose a profiling procedure consisting of the following steps:

1) specifying the target hardware platform and runtime environment (inference runtime);

2) sequential measurement of the execution time of each convolution layer $l \in \{1, ..., L\}$ on a test input tensor with a task-specific dimension;

3) repeating the measurements K times to reduce the impact of random fluctuations and calculating the average latency of each layer;

4) calculating the depthwise latency of each layer.

Let t_l be the measured average latency of the l-th layer, and N_l be the number of output channels of this layer. The depthwise latency is defined as:

$$\delta_l = t_l / N_l , \tag{5}$$

where δ_l characterizes the contribution of a single channel of the l-th layer to the total inference latency, assuming a linear dependence of the layer's latency on the number of its channels.

This linear approximation is acceptable for most convolutional network architectures, since the computational complexity of a convolutional layer is proportional to the number of its output channels when other parameters are fixed (kernel size, number of input channels, spatial dimension). For depth-separated convolutions [13, 14], where the kernel is applied to each channel independently, the linear relationship holds directly.

To ensure comparability of depthwise latencies across different layers, normalization is performed:

$$\tilde{\delta}_l = \frac{\delta_l}{\max_{l'=1..L}\left(\delta_{l'}\right)}, \tag{6}$$

where $\tilde{\delta}_l \in (0,1]$ is the normalized depthwise latency, and the maximum is calculated across all convolutional layers of the network.

4.3 Modified channel importance criterion

The proposed hardware-oriented importance criterion for the j-th channel in the l-th layer is defined as:

$$S_{hw}\left(c_j^l\right) = \frac{S_T\left(c_j^l\right)}{\left(\tilde{\delta}_l\right)^{\beta}}, \tag{7}$$

where $S_T\left(c_j^l\right)$ is the value of the Taylor's basic criterion according to equation (3); is the normalized depthwise latency of the layer according to equation (6); $\beta \in [0,1]$ is a hyperparameter that controls the degree of influence of hardware information on the criterion.

The mechanism of action of the proposed criterion is as follows. Channels with the smallest values of S_{hw} are subject to removal first. Dividing the value of the Taylor criterion by $\left(\tilde{\delta}_l\right)^{\beta}$ leads to a reduction in the importance score of channels in layers with high depthwise latency. Accordingly, when ranking channels by S_{hw}, channels from computationally expensive layers receive lower scores and are more likely to be removed, which directly reduces inference latency.

The hyperparameter β allows adjusting the balance between model accuracy and actual acceleration. When $\beta = 0$, criterion (7) reduces to the standard Taylor criterion (3), i.e., pruning is performed without taking hardware information into account. When $\beta = 1$, the impact of latency is maximized. Intermediate values β ensure a smooth transition between these extreme cases.

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189)
ISSN 3083-7650 (0135-1710) (print) Management Information System and Devices. 2026. No. 2 (189)

4.4 Algorithm for Hardware-Oriented Pruning

The general procedure for applying the proposed criterion consists of the following steps:

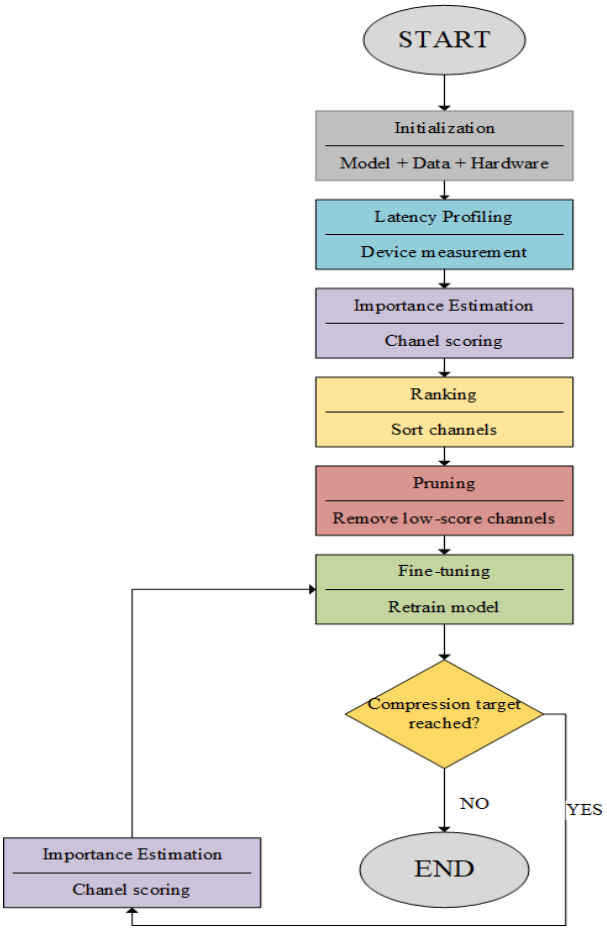

Fig. 1. Block diagram of the hardware-oriented structural pruning algorithm for CNNs

Step 1. Preparation. A pre-trained CNN model and a target hardware platform are selected. A calibration subsample is formed from the training dataset.

Step 2. Latency profiling. For each convolutional layer l, the average latency t_l is measured by repeating the inference K times on the target device. The depthwise latencies δ_l are calculated using formula (5) and their normalized values δ_l are calculated using formula (6).

Step 3. Calculation of importance criteria. A forward and backward pass of the network is performed on the calibration subsample. For each channel c_j^l, the value of the Taylor criterion $S_T\left(c_j^l\right)$ is calculated using formula (3), after which the modified criterion $S_{hw}\left(c_j^l\right)$ is calculated using formula (7) with the selected value β.

Step 4. Ranking and removal. Channels from all network layers are ranked in ascending order by S_{hw}. A specified proportion of channels with the smallest values of S_{hw} is removed. The corresponding filters and BatchNorm parameters are removed from the model.

Step 5. Fine-tuning. The network obtained after pruning is fine-tuned on the full training set for a small number of epochs to restore accuracy.

Step 6. Iteration. If necessary to achieve a higher level of compression, steps 3–5 are repeated iteratively with a gradual increase in the proportion of channels removed.

5. Research Results

5.1 Analysis of the properties of the proposed criterion

The proposed criterion, S_{hw}, has several important properties that determine its behavior in various scenarios.

Property 1 (reduction to the base method). Given $\beta = 0$, for any channel c_j^l, $S_{hw}\left(c_j^l\right) = S_T\left(c_j^l\right)$ holds; that is, the proposed criterion is a generalization of Taylor's criterion. This guarantees that, when the hardware-oriented component is disabled, the pruning results coincide with those of the base method.

Property 2 (priority redistribution between layers). Let two channels c_a and c_b belong to layers l_a and l_b respectively, where $S_T\left(c_a\right) = S_T\left(c_b\right)$, but $\tilde{\delta}_{l_a} > \tilde{\delta}_{l_b}$. Then $S_{hw}\left(c_a\right) < S_{hw}\left(c_b\right)$, i.e., the channel from the computationally more expensive layer receives a lower importance score and will be removed earlier. Under Taylor's standard criterion, both channels would have the same score, and the choice between them would be arbitrary.

Property 3 (monotonicity with respect to β). For a fixed channel c_j^l, the ratio $\dfrac{S_{hw}\left(c_j^l\right)}{S_T\left(c_j^l\right)} = \dfrac{1}{\left(\tilde{\delta}_l\right)^{\beta}}$ monotonically depends on β: it increases as $\tilde{\delta}_l < 1$ and equals one when $\tilde{\delta}_l = 1$ (for a layer with maximum depthwise latency). This means that increasing β intensifies the pruning redistribution in favor of computationally expensive layers.

Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189) *ISSN 3083-7715 (online)*
Management Information System and Devices. 2026. No. 2 (189) *ISSN 3083-7650 (0135-1710) (print)*

5.2 Theoretical Comparison with the Standard Approach

To illustrate the difference between standard and hardware-oriented pruning, let us consider a typical situation that arises when deploying models on edge devices.

Suppose the network contains two convolutional layers: layer A with standard 3×3 convolutions and layer B with depth-separated 3×3 convolutions [14]. With the same number of channels, layer A has significantly higher computational complexity in FLOPs. However, on real hardware, the situation may be different: standard convolutions make more efficient use of GPU/NPU parallelism, whereas depth-split convolutions, despite requiring fewer FLOPs, may execute more slowly due to low utilization of computational units [2].

In this case, the standard Taylor criterion, assuming equal importance of channels in both layers, will remove channels arbitrarily from both layers. The proposed criterion, on the other hand, will automatically identify the layer with higher actual depthwise latency and direct pruning primarily to it, providing greater actual acceleration at the same level of accuracy reduction.

This difference becomes particularly significant for modern detector architectures, such as YOLOv7 [15] or models based on MobileNetV2 [14], which contain layers with heterogeneous operations – standard convolutions, depthwise convolutions, 1×1 pointwise convolutions, concatenations, and element-wise operations. For such networks, the discrepancy between FLOPs and actual latency is most pronounced, and a hardware-oriented approach potentially yields the greatest gains. Experimental validation of this mechanism on a specific detector model with real-world latency profiling is presented in Section 6.

5.3 Analysis of the Hyperparameter's Influence β

The hyperparameter β determines the trade-off between two optimization objectives. At small values of β (close to 0), pruning primarily optimizes model accuracy, as the criterion approaches the standard Taylor criterion. At large values of β (close to 1), pruning more aggressively removes channels from computationally expensive layers, which maximizes actual acceleration but may lead to a greater reduction in accuracy if these channels are critical to the task.

For each "model-hardware platform" pair, there is an optimal value for β^* that provides the best balance between accuracy and latency. β^* can be determined by iterating over a validation sample with a small number of candidate values, such as $\beta \in \{0, 0.25, 0.5, 0.75, 1.0\}$, which is computationally feasible since for each value of β, one only needs to re-rank the channels without re-profiling or computing gradients.

5.4 Computational Complexity of the Proposed Method

The proposed modification introduces minimal additional computational overhead compared to the baseline method. The latency profiling stage is performed once for each target platform and requires $O(K \cdot L)$ direct passes of individual layers, where K is the number of iterations and L is the number of layers. Calculating the modified criterion (7) for all channels requires $O\left(\sum_l N_l\right)$ additional division operations, which is negligibly small compared to the calculation of gradients for the Taylor criterion.

5.5 Comparison of the Proposed Criterion with Existing Approaches

To systematically compare the proposed criterion with existing approaches to structural pruning, six key characteristics have been identified that are critical for the deployment of detection models on edge devices within real-time video systems:

- accounting for the channel's impact on the loss function, which determines the accuracy of assessing the channel's importance for a specific task;
- accounting for the actual latency of the layer on the target hardware, which ensures that pruning is oriented toward actual acceleration rather than a theoretical reduction in FLOPs;
- adaptability to the target platform, i.e., the ability to automatically generate different pruning strategies for different edge devices;
- no need to retrain the model with a modified loss function, which simplifies integration into automated optimization pipelines;
- a mechanism for balancing model accuracy and actual inference acceleration;

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189)
ISSN 3083-7650 (0135-1710) (print) Management Information System and Devices. 2026. No. 2 (189)

- compatibility with existing structural pruning pipelines without changing the training procedure or network architecture.

The comparison results are presented in Table 2.

Analysis of Table 2 allows us to draw the following conclusions. None of the existing criteria takes into account the actual latency of operations on the target hardware platform and does not provide a mechanism for adapting the pruning strategy to a specific edge device. Criteria based on the $L1$ norm [7] and the geometric median [9], despite their computational simplicity and data independence, do not utilize information about the channel's impact on the loss function, which may lead to the removal of channels critical for detection accuracy. The criterion based on BatchNorm [11] only partially accounts for the impact on the loss function – through the $L1$ regularization procedure during training, rather than through direct gradient computation – and requires retraining the network with a modified loss function, which complicates its integration into automated optimization pipelines. The basic Taylor criterion [8] provides the most accurate assessment of channel importance among the approaches considered, but does not distinguish between layers with different computational costs on specific hardware, which can lead to suboptimal acceleration on edge devices.

Table 2. *Comparison of pruning criteria based on characteristics critical for edge deployment*

Characteristic	$L1$-norm	BatchNorm	Taylor	Geometric median	Proposed
Takes into account the impact on the loss function	No	Partially	Yes	No	Yes
Takes into account the actual layer latency	No	No	No	No	Yes
Adapts to the target platform	No	No	No	No	Yes
Does not require model retraining	Yes	No	Yes	Yes	Yes
Adjustable balance between accuracy and speed	No	No	No	No	Yes
Compatibility with existing conveyors	Yes	No	Yes	Yes	Yes

The proposed criterion S_{hw} (7) inherits from Taylor's basic criterion an accurate assessment of a channel's impact on the loss function and supplements it with a hardware-oriented component in the form of a normalized depthwise latency coefficient $\tilde{\delta}_l$. This provides three additional properties absent in all existing approaches: consideration of the actual latency of layers, automatic adaptation to the target platform through profiling, and an adjustable balance between accuracy and speed via the hyperparameter β. At the same time, the proposed criterion remains fully compatible with existing pruning pipelines and does not require model retraining.

6. Experimental verification of the proposed criterion

6.1 Experimental setup

To verify the proposed criterion S_{hw} (Equation 7), the YOLOv8n detector was used in its default configuration – 64 convolutional layers, with an input image size of 640×640 pixels. Latency profiling was performed on a CPU as the most accessible edge computing configuration. For each layer, $K = 50$ measurement repetitions were performed after five warm-up iterations, with the results averaged to minimize the impact of random fluctuations in system call durations. Channel-wise latency δ_l and its normalized value $\tilde{\delta}_l$ were calculated using formulas (5) and (6), respectively. The value of the Taylor criterion S_T (Equation 3) was obtained by performing one forward and one backward pass of the network on the test tensor with a pseudo-functional in the form of the sum of the elements of all output feature maps of the detection heads – this choice of functional ensures the accumulation of gradients on all internal activations of the network. The hyperparameter β in formula (7) is set to 0.5, which corresponds to a balanced pruning redistribution mode.

6.2 Layer Latency Profile and Hardware Heterogeneity of the Model

Fig. 2(a) shows the absolute latency of each convolutional layer of the YOLOv8n model. The distribution is noticeably heterogeneous: values range from 0.009 ms to 0.135 ms, i.e., by more than one and a half orders of magnitude. The most computationally expensive operations are the convolutions 3×3 in the middle part of the network (indices 20–24, 41–44) and the stream convolutions 1×1 that operate on high-resolution maps (indices 22, 25–26, 41–42). In Fig. 2(b), the same profile is presented in the form of normalized depthwise latency $\tilde{\delta}_l$, which is directly included in formula (7). One of the detection head layers (index ≈ 65)

Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189) *ISSN 3083-7715 (online)*
Management Information System and Devices. 2026. No. 2 (189) *ISSN 3083-7650 (0135-1710) (print)*

exhibits $\tilde{\delta}_l = 1{,}0$, meaning its contribution to the depthwise inference time is tens of times greater than that of most other layers. The obtained profile experimentally confirms the initial hypothesis of this work: the hardware cost of channels is significantly heterogeneous, and the uniform treatment of channels, characteristic of the classical Taylor criterion, ignores this heterogeneity.

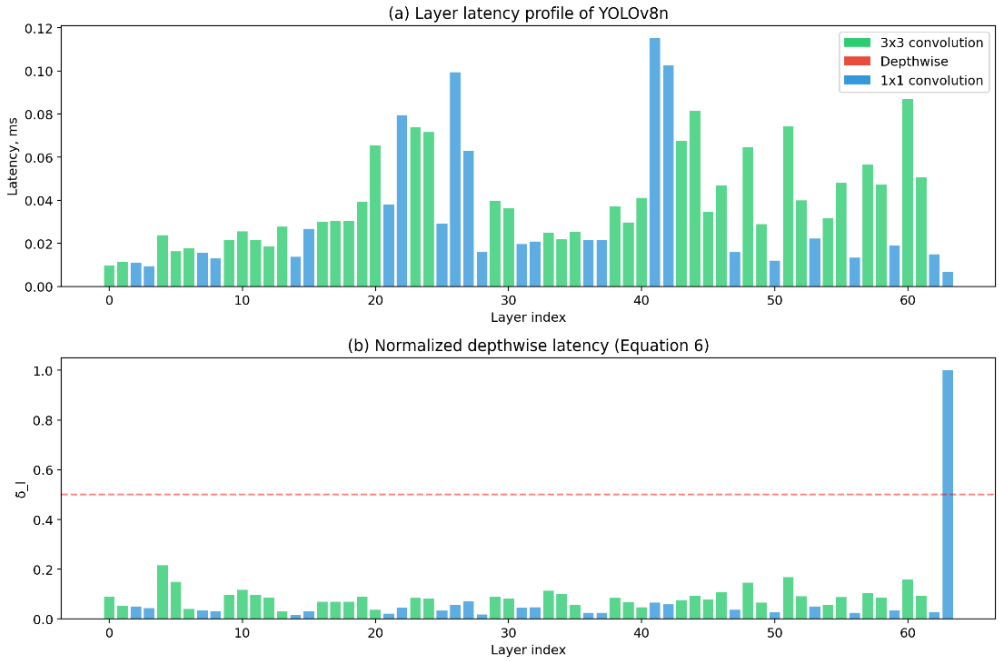

Fig. 2. Variation in the computational cost of the convolutional layers of the YOLOv8n model: (a) absolute layer latency, ms; (b) normalized depthwise latency (Equation 6)

6.3 Pruning Redistribution and Quantitative Advantage of the Proposed Criterion

Fig. 3(a) presents normalized estimates of layer importance based on two criteria. The distribution shapes are similar, but the differences are concentrated precisely in the regions that have the greatest impact on inference time: for layers with low $\tilde{\delta}_l$, the estimates S_{hw} decrease relative to S_T, while for layers with high $\tilde{\delta}_l$, they increase. This constitutes the methodological core of the proposed approach: the importance of a channel is normalized by the hardware cost of the channel, as a result of which "cheap" channels are prioritized for retention, and "expensive" ones for removal.

Fig. 3(b) illustrates the result of applying this redistribution – the proportion of removed channels in each layer with a total pruning budget of 30%. As expected, both criteria do not affect the early layers of the backbone (indices 0–25), which have the highest importance score and are responsible for low-level features. However, in the middle and deep parts of the network (indexes 31–45), a systematic shift is observed: S_{hw} prunes slower convolutions more aggressively 3×3 and preserves "cheap" ones 1×1, whereas the standard approach distributes removals without accounting for this difference.

Quantitative comparison results are presented in Table 3. With the same number of removed channels (1,718, corresponding to 30% of the total number of model channels), the proposed criterion achieved a projected latency reduction of 1.0727 ms versus 0.9705 ms for the standard Taylor criterion, or 38.55% and 34.88%, respectively, of the model's baseline latency. In relative terms, this represents a +10.5% additional performance gain achieved solely through hardware-oriented pruning redistribution, without changing the number of retained parameters and without additional overhead at the importance evaluation stage ($\tilde{\delta}_l$ profiling is performed once). The number of layers in which pruning exceeds 5% increases slightly (33 versus 31), indicating a more balanced distribution of removals: S_{hw} avoids concentrating pruning in the few fastest layers and distributes it more broadly, but purposefully – toward the slower ones.

ISSN 3083-7715 (online)
ISSN 3083-7650 (0135-1710) (print)
Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189)
Management Information System and Devices. 2026. No. 2 (189)

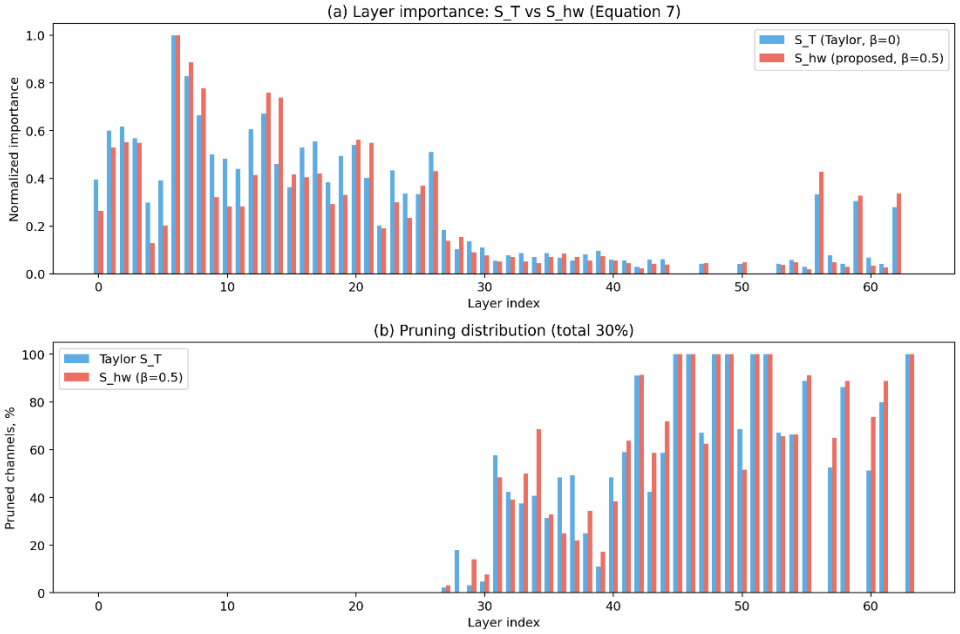

Fig. 3. Effect of the hardware-oriented component of the criterion (Formula 7) on the YOLOv8n pruning plan:
a) redistribution of normalized layer importance between S_T and S_{hw};
b) proportion of removed channels by layer with a 30% pruning budget.

Table 3. *Comparison of pruning criteria for S_T and S_{hw} for the YOLOv8n model with a 30% pruning budget (CPU)*

Metric	**Taylor** S_T $(\beta = 0)$	S_{hw} $(\beta = 0.5)$
Channels removed	1718	1718
Layers with pruning > 5%	31	33
Maximum pruning of a single layer, %	100	100
Latency reduction, ms	0.9705	1.0727
Latency reduction, %	34.88	38.55
Advantage over the Taylor method	–	+10.5%

Note that the latency reduction shown in Table 3 is theoretical – based on the assumption of a linear relationship between a layer's inference time and the number of its channels, as given by Equation (5). The physical implementation of pruning, together with full model retraining, will provide final empirical verification of the preservation of detection quality and actual latency, which is a direct focus of future research.

7. Discussion of the research results

The proposed hardware-oriented criterion for structural pruning has several practical advantages for model deployment tasks in video systems with local data processing. Let us consider these advantages using specific examples.

7.1 Automatic detection of hardware "bottlenecks"

Criterion (7) automatically identifies layers with disproportionately high actual latency, even if their theoretical complexity in FLOPs is low. For example, when deploying MobileNetV2-SSD on an ARM Cortex-A53, depth-separated convolutions have significantly fewer FLOPs compared to standard ones, yet they execute inefficiently due to low arithmetic intensity. Profiling will reveal high depthwise latency $\tilde{\delta}_l$ in these layers, and criterion (7) will lower the importance score of their channels, directing pruning specifically at them. The standard Taylor criterion lacks a mechanism for detecting such situations.

Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189) *ISSN 3083-7715 (online)*
Management Information System and Devices. 2026. No. 2 (189) *ISSN 3083-7650 (0135-1710) (print)*

7.2 Adaptability to different hardware platforms

Since the coefficient $\tilde{\delta}_l$ is calculated by profiling a specific device, the same algorithm automatically generates different pruning strategies for different platforms. For example, for YOLOv7-tiny, the convolutions 3×3 on the NVIDIA Jetson Nano effectively utilize the parallelism of CUDA cores, whereas on the Google Coral Edge TPU, those same layers may be partially executed on the host device's CPU. Different latency profiles will result in different channel rankings based on criterion (7), which is optimal for each platform. The standard Taylor criterion will yield the same ranking regardless of the device.

7.3 Modularity and minimal additional costs

The modification does not require changes to the training procedure, network architecture, or pruning algorithm. Integration into any existing pipeline with depthwise ranking boils down to a one-time platform profiling and element-by-element decomposition of the Taylor criterion vector into a vector of normalized latencies according to formula (7), which requires $O\left(\sum_l N_l\right)$ additional operations – negligibly few compared to gradient computation.

In the context of real-time video systems, the proposed method is particularly relevant, as inference latency requirements are stringent and determined by the video stream's frame rate. A typical video surveillance task requires processing 25–30 frames per second, corresponding to a latency budget of 33–40 ms per frame. Traditional pruning, focused on minimizing FLOPs, may fail to meet this budget even with a significant reduction in computational complexity if the "bottleneck" consists of layers with inefficient execution on specific hardware.

It is important to note the limitations of the proposed approach. The linear approximation of the relationship between layer latency and the number of channels (5) may be inaccurate for some configurations, particularly for layers with a very small number of channels, where the overhead of launching the computation kernel dominates the useful computations. In addition, the latency profile may vary depending on the driver version, the device's operating temperature, and other external factors.

The experimental validation performed in Section 6 using the YOLOv8n model confirmed the main hypothesis of this work: hardware-oriented channel ranking provides a 10.5% reduction in latency for the same budget of removed channels, and this advantage is achieved without additional overhead during the importance evaluation stage. Note that the experiment was conducted on a CPU as the most accessible edge configuration; extending the verification to specialized accelerators – NVIDIA Jetson Nano (CUDA-core parallelism), Google Coral Edge TPU (quantized INT8 arithmetic), and the ARM Cortex-A series (no GPU acceleration) – constitutes the immediate focus of further research, since it is on such platforms that $\tilde{\delta}_l$ profiles are expected to be the most heterogeneous. Separate research directions include the study of network architectures with a significant proportion of depthwise operations (MobileNetV2-SSD, EfficientDet-Lite), rigorous verification of the preservation of detection performance after the physical implementation of pruning according to both criteria, as well as combining the proposed pruning with subsequent quantization to achieve a cumulative acceleration effect.

8. Conclusion

This paper presents a comparative analysis of existing criteria for structural pruning of convolutional neural networks – based on the $L1$ norm, batch normalization scaling factors, the Taylor series, and the geometric median. The choice of the Taylor criterion as the baseline method is justified due to its theoretical soundness and ability to account for the actual influence of the channel on the loss function.

An improvement to the Taylor criterion is proposed by introducing a normalized depthwise latency coefficient, calculated based on profiling the execution time of each layer on the target edge device. The modified criterion ensures a redistribution of pruning in favor of layers with high actual computational cost, which allows for greater inference acceleration compared to the hardware-independent approach with controlled loss of accuracy.

A theoretical analysis of the properties of the proposed criterion was performed; in particular, its reduction to the baseline method was shown for $\beta = 0$, the monotonicity of priority reallocation with respect to the hyperparameter β, and the minimal additional computational complexity.

Experimental verification on a YOLOv8n detector model with 64 convolutional layers confirmed the

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189)
ISSN 3083-7650 (0135-1710) (print) Management Information System and Devices. 2026. No. 2 (189)

theoretical predictions. With a pruning budget of 30% of removed channels, the modified criterion achieved a projected latency reduction of 38.55% compared to 34.88% for the baseline Taylor criterion, corresponding to a relative advantage of +10.5%. An analysis of the pruning distribution across layers showed that the achieved advantage is due precisely to the targeted removal of channels in layers with high depthwise latency $\left(\tilde{\delta}_l\right)$, which directly confirms the functioning of the mechanism described in Equation (7).

The proposed approach is modular and can be integrated into existing model optimization pipelines for video systems with local data processing. Further research is aimed at extending experimental verification to specialized edge accelerators and architectures with a higher proportion of depthwise operations, as well as investigating the combination of the proposed pruning with subsequent quantization to achieve a cumulative acceleration effect.

Conflict of Interest

The authors declare that they have no conflicts of interest regarding this study, including financial, personal, authorship, or other conflicts that could influence the study and its results presented in this article.

Funding

The study was conducted without financial support.

Data Availability

The manuscript has no associated data.

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence technologies in the creation of this work.

References

1. Wang, X., Han, Y., Leung, V. C. M., Niyato, D., Yan, X., Chen, X. (2023), "Edge computing with artificial intelligence: A machine learning perspective", *ACM Computing Surveys*, Vol. 55, No. 9, Article 184. DOI: https://doi.org/10.1145/3555802
2. Yang, Z., Hu, J., Tang, X., Zhao, L., Li, K. (2022), "Optimization methods, challenges, and opportunities for edge inference", *Electronics*, Vol. 11, No. 14, Article 2189. DOI: https://doi.org/10.3390/electronics11142189
3. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D. (2018), "Quantization and training of neural networks for efficient integer-arithmetic-only inference", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: https://doi.org/10.1109/CVPR.2018.00286
4. Li, Z., Xu, P., Chang, X., Yang, L., Zhang, Y., Yao, L., Chen, X. (2023), "When object detection meets knowledge distillation: A survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*. DOI: https://doi.org/10.1109/TPAMI.2023.3257546
5. Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S. (2022), "A ConvNet for the 2020s", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: https://doi.org/10.1109/CVPR52688.2022.01167
6. Cheng, H., Zhang, M., Shi, J. Q. (2024), "A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*. DOI: https://doi.org/10.1109/TPAMI.2024.3447085
7. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H. P. (2017), "Pruning filters for efficient convnets", *Proceedings of the International Conference on Learning Representations (ICLR)*. DOI: https://doi.org/10.48550/arXiv.1608.08710
8. Molchanov, P., Mallya, A., Tyree, S., Frosio, I., Kautz, J. (2019), "Importance estimation for neural network pruning", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: https://doi.org/10.1109/CVPR.2019.01161
9. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y. (2019), "Filter pruning via geometric median for deep convolutional neural networks acceleration", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: https://doi.org/10.1109/CVPR.2019.00447
10. Fang, G., Ma, X., Song, M., Mi, M. B., Wang, X. (2023), "DepGraph: Towards any structural pruning", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: https://doi.org/10.1109/CVPR52729.2023.01544

Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189) *ISSN 3083-7715 (online)*
Management Information System and Devices. 2026. No. 2 (189) *ISSN 3083-7650 (0135-1710) (print)*

11. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C. (2017), "Learning efficient convolutional networks through network slimming", *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. DOI: https://doi.org/10.1109/ICCV.2017.298
12. He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., Han, S. (2018), "AMC: AutoML for model compression and acceleration on mobile devices", *Proceedings of the European Conference on Computer Vision (ECCV)*. DOI: https://doi.org/10.1007/978-3-030-01234-2_48
13. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. (2017), "MobileNets: Efficient convolutional neural networks for mobile vision applications", *arXiv preprint*. DOI: https://doi.org/10.48550/arXiv.1704.04861
14. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C. (2018), "MobileNetV2: Inverted residuals and linear bottlenecks", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: https://doi.org/10.1109/CVPR.2018.00474
15. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y. M. (2023), "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: https://doi.org/10.1109/CVPR52729.2023.00721

Received (Надійшла) 18.03.2026
Accepted for publication (Прийнята до друку) 07.05.2026
Publication date (Дата публікації) 29.05.2026

Відомості про авторів / About the Authors

Нечипуренко Станіслав Ігорович – Харківський національний університет радіоелектроніки, аспірант кафедри електронних обчислювальних машин; Харків, Україна;
Stanislav Nechypurenko – Kharkiv National University of Radio Electronics, Postgraduate Student, Department of Electronic Computers; Kharkiv, Ukraine;
e-mail: stanislav.nechypurenko@nure.ua
ORCID ID: https://orcid.org/0009-0007-2936-9935

Сорокін Антон Романович – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, доцент кафедри електронних обчислювальних машин; Харків, Україна;
Anton Sorokin – Phd (Technical Sciences), Associate Professor, Kharkiv National University of Radio Electronics, Associate Professor Department of Electronic Computers; Kharkiv, Ukraine;
e-mail: anton.sorokin@nure.ua
ORCID ID: https://orcid.org/0000-0002-4383-2611

МЕТОД СТРУКТУРНОГО ПРУНІНГУ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ З ОГЛЯДУ НА АПАРАТНІ ХАРАКТЕРИСТИКИ EDGE-ПРИСТРОЇВ

Предмет дослідження – методи структурного прунінгу згорткових нейронних мереж, що застосовуються для зменшення обчислювальної складності моделей детекції об’єктів у відеосистемах з локальним обробленням даних. **Мета роботи** – вдосконалення методу структурного прунінгу способом упровадження апаратно-орієнтованого критерію відбору каналів, який зважає на реальну латентність операцій на цільовому edge-пристрої. **Завдання дослідження**: порівняльний аналіз наявних критеріїв структурного прунінгу, обґрунтування вибору базового методу й розроблення його модифікації на основі профілювання латентності цільової апаратної платформи. **Методи дослідження** основані на математичному аналізі критеріїв важливості каналів згорткових шарів, теоретичному порівнянні підходів до оцінювання надлишковості параметрів нейронної мережі та формалізації задачі апаратно-орієнтованої оптимізації. **Результати дослідження** полягають у розробленні модифікованого критерію важливості каналів, що поєднує оцінку впливу каналу на функцію втрат із нормалізованим коефіцієнтом латентності відповідного шару, який обчислюється способом профілювання часу виконання кожного згорткового шару на цільовому edge-пристрої. Запропонований критерій дає змогу пріоритизувати вилучення каналів у шарах, що створюють найбільше навантаження на конкретному апаратному забезпеченні, водночас зберігаючи канали

ISSN 3083-7715 (online) *Автоматизовані системи управління та прилади автоматики. 2026. № 2 (189)*
ISSN 3083-7650 (0135-1710) (print) *Management Information System and Devices. 2026. No. 2 (189)*

з високим впливом на точність моделі. Виконано теоретичний аналіз властивостей критерію, зокрема доведено його зведення до базового критерію Тейлора за умови нульового значення гіперпараметра та монотонність перерозподілу пріоритетів прунінгу між шарами. Продемонстровано, що запропонований підхід є модульним, потребує мінімальних додаткових обчислювальних витрат і може бути інтегрований у наявні конвеєри оптимізації моделей. Експериментальна перевірка на моделі детектора YOLOv8n підтвердила запропонований підхід: за бюджету прунінгу 30% вилучених каналів отримано зниження латентності інференсу на 38,55% проти 34,88% у базового критерію Тейлора, що відповідає відносній перевазі +10,5%. **Висновки.** Результати підтверджують теоретичну обґрунтованість запропонованого підходу та його переваги порівняно з апаратно-незалежними критеріями прунінгу для задач розгортання моделей детекції на edge-пристроях у складі відеосистем реального часу.

Ключові слова: структурний прунінг; згорткова нейронна мережа; edge-пристрій; латентність інференсу; критерій важливості каналів; оптимізація моделі; відеоаналітика; детекція об’єктів; локальне оброблення даних; апаратно-орієнтована оптимізація.

Бібліографічні описи / Bibliographic descriptions

Нечипуренко С. І., Сорокін А. Р. Метод структурного прунінгу згорткових нейронних мереж з огляду на апаратні характеристики edge-пристроїв. *Автоматизовані системи управління та прилади автоматики*. 2026. № 2 (189). С. 34–46. DOI: https://doi.org/10.30837/0135-1710.2026.189.034

Nechypurenko, S., Sorokin, A. (2026), "Method of structural pruning of convolutional neural networks with regard to edge device hardware characteristics", *Management Information System and Devices*, No. 2 (189), P. 34–46. DOI: https://doi.org/10.30837/0135-1710.2026.189.034